



Escola d'Enginyeria de Telecomunicació i  
Aeroespacial de Castelldefels

UNIVERSITAT POLITÈCNICA DE CATALUNYA

# TRABAJO FINAL DE GRADO

**TÍTULO DEL TFG:** Diseño de un módulo de cuestionarios para una aplicación de gamificación.

**TITULACIÓN:** Grado en Ingeniería Telemática

**AUTOR:** Rubèn Cobos Bermúdez

**DIRECTORES:** Miguel Valero García  
Roc Meseguer Pallarès

**FECHA:** 17 de Junio del 2020



**Título:** Diseño de un módulo de cuestionarios para una aplicación de gamificación.

**Autor:** Rubèn Cobos Bermúdez

**Directores:** Miguel Valero García  
Roc Meseguer Pallarès

**Fecha:** 17 de Junio del 2020

## Resumen

El proyecto que veremos a continuación está basado en el diseño e implementación de un módulo de cuestionarios para una aplicación de gamificación en la docencia, llamada Classpip. Esta herramienta hace uso del concepto de gamificación, el cual consiste en usar técnicas propias de los juegos (competición, recompensas...), para potenciar la motivación y el aprendizaje en entornos educativos.

En la actualidad, Classpip está formado por 4 proyectos entrelazados: Classpip-services que es la API y está desarrollada con *LoopBack*, Classpip-dashboard que es una aplicación web desarrollada con *Angular*, Mobile-Profesor y Mobile-Alumno las cuales son las aplicaciones móviles para el rol de profesor y de alumno respectivamente y están desarrolladas con *Ionic*.

El objetivo del proyecto es implementar un módulo de cuestionarios, el cual he desarrollado en Services, Dashboard y el Mobile-Alumno. Empezando por Services, donde he puesto en funcionamiento todos los modelos de datos necesarios. Seguidamente en el Dashboard, el cual he dividido en tres partes: La primera, el módulo de preguntas que gestiona toda la creación y edición de las mismas, segundo el de cuestionarios que se basa también en su correspondiente creación y edición haciendo uso de las preguntas creadas anteriormente, y para acabar el módulo de juego de cuestionario que contiene todas las características necesarias para poder crear y gestionar el juego. Por otro lado, en el Mobile-Alumno he implementado el juego de cuestionario para que el alumno pueda disfrutar de él y pueda jugar a lo que haya creado el profesor anteriormente.

Finalmente, en la memoria podemos ver detalladamente todo el proceso de creación del módulo, las diferentes decisiones que se han ido tomando a lo largo del proceso de desarrollo, también veremos cómo está estructurado el código y las diferentes pruebas que se tienen que hacer a la hora de probar el módulo. Para acabar veremos unas conclusiones sobre cómo ha ido el proyecto en general y que me ha aportado personalmente.

**Title:** Design of a questionnaire module for a gamification app

**Author:** Rubèn Cobos Bermúdez

**Directors:** Miguel Valero García  
Roc Meseguer Pallarès

**Date:** June 17<sup>th</sup>, 2020

## Overview

The project that we will see below is based on the design and implementation of a module of questionnaires for a gamification application in teaching, called Classpip. This tool uses the concept of gamification, which consists in using techniques specific to games (competition, rewards...), to enhance motivation and learning in educational environments.

Classpip currently consists of 4 interlocking projects: Classpip-services which is the API and is developed with *Loopback*, Classpip-dashboard which is a web application developed with *Angular*, Mobile-Profesor and Mobile-Alumno which are mobile applications for the role of teacher and student respectively and are developed with *Ionic*.

The objective of the project is to implement a module of questionnaires, which I have developed in Services, Dashboard and Mobile-Student. Starting with Services, where I implemented all the necessary data models. Then, in the Dashboard, which I have divided into three parts: first, the module of questions that manages all the creation and editing of the questions that the teacher creates, second the one of questionnaires which is also based on its corresponding creation and editing making use of the questions that have been created previously, and the last one the module of quiz game that contains all the necessary features to be able to create and manage the game. On the other hand, in the Mobile-Student I have implemented the quiz game so that the student can enjoy it and can play what the teacher created previously.

Finally, in the report we can see in detail the whole process of creating the module, the different decisions that have been taken throughout the development process, we will also see how the code is structured and the different tests that have to be done to test the module. Finally, we will see some conclusions about how the project in general has gone and that it has brought me personally.

## **Agradecimientos**

Gracias a mis tutores Miguel y Roc,  
por haberme ayudado en todo lo posible y por estar siempre pendientes.

A mis padres, hermana y a mis abuelos,  
gracias por no tirar nunca la toalla conmigo  
y por apoyarme en todas y cada una de mis decisiones.  
De no ser por vosotros no habría llegado aquí, Gracias.

# ÍNDICE

<b>CAPÍTULO 1. INTRODUCCIÓN .....</b>	<b>1</b>
1.1 Módulo de cuestionarios .....	1
1.2 Proyecto Classpip .....	1
1.3 Estructura de la memoria .....	2
<b>CAPÍTULO 2. GAMIFICACIÓN .....</b>	<b>4</b>
2.1 ¿Qué es la Gamificación? .....	4
2.2 Herramientas de uso en la educación .....	5
2.2.1 Plickers .....	5
2.2.2 Quizlet .....	5
2.2.3 Flipquiz .....	6
2.2.4 Socrative .....	7
2.3 Lluvia de ideas .....	7
<b>CAPÍTULO 3. INTRODUCCIÓN A CLASSPIP .....</b>	<b>9</b>
3.1 Descripción del proyecto Classpip .....	9
3.1.1 Los juegos de Classpip .....	9
3.1.2 Colecciones .....	11
3.1.3 Puntos e Insignias .....	11
3.1.4 Alumnos, Grupos y Equipos .....	12
3.2 Arquitectura del proyecto .....	12
<b>CAPÍTULO 4. OBJETIVOS Y PLAN DE TRABAJO .....</b>	<b>14</b>
4.1 Objetivos .....	14
4.2 Plan de trabajo .....	15
<b>CAPÍTULO 5. DISEÑO FUNCIONAL .....</b>	<b>16</b>
5.1 Módulo de cuestionarios .....	16
5.1.1 Mecánica del juego .....	16
5.1.2 Las preguntas .....	16
5.1.3 Los cuestionarios .....	17
5.1.4 El juego de cuestionario .....	17
5.2 Funciones a realizar desde el Dashboard .....	18
5.3 Funciones a realizar desde Mobile-Alumno .....	18
<b>CAPÍTULO 6. DISEÑO DE PANTALLAS .....</b>	<b>20</b>
6.1 Mockups Dashboard .....	20
6.2 Mockups Mobile-Alumno .....	24

6.3	Estilos gráficos .....	26
<b>CAPÍTULO 7. MODELO DE DATOS E IMPLEMENTACIÓN .....</b>		<b>28</b>
7.1	Modelo de datos .....	28
7.2	Implementación modelo de datos.....	30
7.2.1	Preguntas .....	31
7.2.2	Cuestionario.....	31
7.2.3	Pregunta del cuestionario .....	32
7.2.4	Juego de cuestionario .....	33
7.2.5	Alumno juego de cuestionario .....	33
7.2.6	Respuesta alumno juego de cuestionario .....	34
7.3	Consultas a la base de datos .....	34
7.3.1	Ejemplo POST .....	35
7.3.2	Ejemplo GET .....	36
7.3.3	Ejemplo PUT.....	36
7.3.4	Ejemplo DELETE .....	37
<b>CAPÍTULO 8. IMPLEMENTACIÓN APLICACIONES .....</b>		<b>39</b>
8.1	Dashboard y móvil del alumno. ....	39
8.1.1	Modelos .....	39
8.1.2	Componentes .....	40
8.1.3	Servicios .....	41
8.2	Cuantificación trabajo realizado .....	43
8.3	Algoritmo crítico .....	43
8.3.1	Ordenación de preguntas y respuestas .....	43
8.3.1.1	Preguntas y respuestas mismo orden.....	44
8.3.1.2	Preguntas desordenadas .....	45
8.3.1.3	Preguntas y respuestas desordenadas .....	46
<b>CAPÍTULO 9. PRUEBAS.....</b>		<b>47</b>
9.1	Lista de pruebas .....	47
9.1.1	Pruebas módulo de preguntas .....	47
9.1.2	Pruebas módulo de cuestionario .....	47
9.1.3	Pruebas módulo juego de cuestionario .....	48
<b>CAPÍTULO 10. EVALUACIÓN .....</b>		<b>49</b>
10.1	Modelo de evaluación .....	49
10.2	Evaluación de las respuestas .....	53
<b>CAPÍTULO 11. CONCLUSIONES .....</b>		<b>54</b>
11.1	Conclusiones técnicas.....	54
11.2	Conclusiones personales .....	55
11.3	Propuestas para futuras mejoras .....	55

<b>ANEXOS .....</b>	<b>57</b>
<b>ANEXO 1 Resultado final .....</b>	<b>57</b>
Pantallas Dashboard .....	57
Módulo de preguntas .....	57
Módulo de cuestionarios .....	59
Módulo juego de cuestionario .....	61
Pantallas Mobile-Alumno .....	63
Módulo juego de cuestionario .....	63
<b>ANEXO 2 Estilos gráficos .....</b>	<b>65</b>
Estilos gráficos Dashboard .....	65
Títulos .....	65
Botones .....	65
Tablas .....	66
Módulo de creación .....	66
Alertas .....	67
Estilos gráficos Mobile-Alumno .....	67
Botones .....	67
Tablas .....	68
Módulo de juego .....	68
Alertas .....	68



## CAPÍTULO 1. INTRODUCCIÓN

En este capítulo vamos a ver las ideas principales del proyecto. Primero, veremos de que trata el proyecto, después en que consiste Classpip y seguidamente veremos cómo está estructurada la memoria en sí.

### 1.1 Módulo de cuestionarios

En este proyecto he desarrollado un módulo de cuestionarios para Classpip (que es una herramienta de gamificación en la docencia que hace uso de las mecánicas y dinámicas de los juegos, la cual veremos en detalle posteriormente). Este módulo nos permite, como su propio nombre indica, crear cuestionarios de una forma lúdica. Estos, estarán creados por el profesor, de principio a fin, empezando por la creación de las preguntas y acabando por el juego en sí, y les podrá dar uso en el momento que le parezca más adecuado. En ese momento los alumnos desde su dispositivo móvil tendrán que contestarlos dentro del tiempo que les estipulen y cuando lo finalicen recibirán el resultado que han adquirido en el cuestionario que hayan realizado. Este módulo, permite que los cuestionarios se adapten a las necesidades tanto del profesor como del alumno ya que es completamente editable y no hay nada predefinido. Por otro lado, he intentado que sea lo más intuitivo posible y con un gran abanico de posibilidades a la hora de crearlos, para intentar hacerle sencillo al maestro el uso de esta herramienta y a los alumnos a la hora de contestarlo.

Mi objetivo ha sido el de aportar mi granito de arena a este gran proyecto que es Classpip y añadir un módulo lo más funcional posible, mejorando de esta forma la aplicación y dándole más valor del que ya tiene. La idea principal es la creación de una herramienta para que los alumnos puedan aprender de una forma más dinámica y divertida posible, pero sin perder la importancia que tiene siempre un cuestionario, de tal forma que tanto el alumno como el profesor puedan ver cómo van en un momento determinado del curso o simplemente usarlo como herramienta de aprendizaje.

Dicho esto, vamos a ver en que consiste Classpip.

### 1.2 Proyecto Classpip

A día de hoy, la tecnología es una parte fundamental en del día a día de las personas y forma parte dentro de todos los sectores (educativo, farmacéutico, construcción, ...). Si nos adentramos dentro del sector educativo podemos apreciar que, en muchos casos, se dice que la tecnología es un atraso para los alumnos y no un avance. ¿Pero y si consiguiéramos que la tecnología fuera un aliciente o mejora para que los alumnos puedan mejorar su rendimiento y atención en clase? Bien, Classpip mediante el uso de la tecnología y de la metodología de la gamificación, quiere conseguir este objetivo. La gamificación aporta los mecanismos y técnicas que encontramos en los juegos en escenarios que no son juegos, para que los alumnos aprendan y a la vez se diviertan haciendo cosas que quizás no tienen muchas ganas de hacer. Gracias a ello,

conseguimos que los alumnos tengan una mayor motivación a la hora de atender en clase.

Este proyecto, fue creado por el Departamento de Arquitectura de Computadores de la Universidad Politécnica de Cataluña (UPC), la arquitectura de la aplicación fue desarrollada en un TFM (Trabajo final de master) por un alumno en 2016 y ha seguido progresando gracias a la ayuda de diferentes alumnos añadiendo funcionalidades a Classpip en sus TFC (Trabajo final de carrera). En la aplicación, por un lado, tenemos la visión del profesor que utilizará la aplicación desde su ordenador (WebApp) o dispositivo móvil, y por el otro, el alumno que lo utilizará desde el dispositivo que tenga disponible (móvil o Tablet).

Classpip actualmente está dividido en cuatro grandes proyectos: Classpip-services, Classpip-dashboard, Mobile-Alumno y Mobile-Profesor. Conectados cada uno entre sí, haciendo que la aplicación sea lo más útil y dinámica posible. Cada uno de estos proyectos está enfocado a un perfil distinto de usuario, dependiendo del rol que tenga estipulado, ya sea profesor o alumno.

Empezando por el Dashboard, se puede decir que es la base de operaciones, ya que desde aquí el profesor tiene acceso a las diferentes funcionalidades de la herramienta, por ejemplo, gestión de las listas de alumnos o grupos, creación de juegos, asignación de puntos e insignias en los juegos de puntos y también permite crear y asignar álbumes de cromos en los juegos de colección. Todas y cada una de estas funcionalidades no podrían llevarse a cabo de no ser por el proyecto de servicios (Classpip-services), ya que es la API a través de la cual se lanzan todas las peticiones de datos. Todos estos datos, por el momento, se almacenan en un fichero que hace la función de “base de datos”, pero posteriormente se acabaran guardando dentro de una base de datos real.

Por último, tenemos la aplicación del móvil del profesor y del alumno (Mobile-Alumno y Mobile-Profesor), en las cuales no se encuentran implementadas todas las funcionalidades que tenemos disponibles en el Dashboard, ya que no todas son adecuadas para implementar en un dispositivo móvil (por ejemplo, en el caso de añadir una imagen en la creación del perfil de un alumno es más cómodo y sencillo realizarlo desde el Dashboard que desde el dispositivo móvil) y también por el simple hecho de que las aplicaciones móviles están planteadas para dar soporte a la aplicación web. Por un lado, el móvil del profesor está enfocado como una herramienta de consulta, mientras que el móvil del alumno, está más enfocado a poder interactuar con los diferentes juegos y para darle al alumno un papel importante dentro de Classpip.

## 1.3 Estructura de la memoria

Después de ver en que consiste el proyecto y que es Classpip, vamos a ver cómo está estructurada la memoria:

- El siguiente capítulo es una explicación sobre que es la Gamificación y veremos unos cuantos ejemplos de aplicaciones que hacen uso del

módulo de cuestionarios. Gracias a esta información, lo siguiente será una lluvia de ideas sobre cómo puedo realizar el módulo (**capítulo 2**).

- Después en el **capítulo 3** veremos qué es Classpip, sobre cómo está estructurado el proyecto, y sobre los diferentes juegos y módulos que tiene implementados.
- En el siguiente (**capítulo 4**) veremos los objetivos de este proyecto y el plan de trabajo a seguir para el desarrollo del mismo.
- En el **capítulo 5** será el punto de partida del desarrollo del módulo. En este apartado analizaremos en qué va a consistir el módulo de cuestionarios y las diferentes partes que tendrá.
- En el **capítulo 6** podemos ver los diferentes diseños de la app (pantallas) que he creado y como va a ser la transición de una página a otra, para poder tener una idea más visual del módulo en sí, e incluso veremos las diferentes funcionalidades que tendrá cada una de ellas.
- En el siguiente (**capítulo 7**) hablaremos sobre la base de datos que hay planteada para el módulo y posteriormente veremos su implementación.
- En el **capítulo 8** veremos como he desarrollado el módulo y su estructura, para comprender mejor el funcionamiento. Por otro lado, veremos una serie de datos que nos ayudaran a ver la magnitud de trabajo que ha tenido este proyecto. Y para finalizar veremos el algoritmo más complejo utilizado junto con su correspondiente explicación, para que se pueda entender mejor.
- El siguiente capítulo (**capítulo 9**) habla de las diferentes pruebas que hay que ejecutar para poder comprobar que la aplicación funciona correctamente y que no tiene ningún error.
- En el **capítulo 10** tenemos una evaluación del proyecto para la cual me han ayudado una serie de personas probando la herramienta y posteriormente realizando un formulario de evaluación, para de esta forma sacar unas mejores conclusiones.
- Seguidamente, en el **capítulo 11** analizaremos el resultado final en unas conclusiones tanto técnicas como personales del proyecto. Por último, expondré una serie de posibles ideas de mejora para futuras versiones.
- Para finalizar, tenemos los anexos del proyecto, en el primero (**Anexo 1**) veremos el resultado final del módulo tanto en el Dashboard como en el móvil del alumno. Segundo, en el **Anexo 2** veremos los diferentes estilos gráficos que he usado en el proyecto.

## CAPÍTULO 2. GAMIFICACIÓN

En este capítulo vamos a hablar sobre la Gamificación y veremos en qué consiste, también analizaremos unos cuantos ejemplos de herramientas que hacen uso de ello, y por último veremos en la lluvia de ideas para el proyecto, que ha surgido a raíz del estudio realizado.

### 2.1 ¿Qué es la Gamificación?

La Gamificación es una técnica de aprendizaje que traslada la mecánica de los juegos al ámbito educativo-profesional. Esta técnica, se está empezado a utilizar en muchos centros, con el fin de conseguir mejores resultados, ya sea para mejorar la concentración, para intentar absorber una mayor cantidad de conocimientos, o bien para recompensar acciones concretas, entre otros muchos objetivos.

Esta metodología está ganando enteros en el ámbito educativo y formativo, ya que, gracias a su carácter lúdico, facilita la absorción de conocimientos de una forma más divertida, generando en el usuario una experiencia muy positiva.

El modelo de juego genera mucha motivación en el alumno, saca a relucir su ánimo de superación y por lo tanto esto genera un mayor compromiso. Si junto con la actitud del alumno le sumamos un conjunto técnicas mecánicas (recompensas) y dinámicas (motivación) extrapoladas de los juegos, llegaríamos a obtener unos mejores resultados. Algunas de las técnicas mecánicas más utilizadas son las siguientes:

- **Acumulación de puntos:** Se le otorga un valor numérico a cada una de las acciones que realiza el alumno y se van acumulando.
- **Misiones o retos:** Conseguir superar un reto o un objetivo planteado, ya sea solo o en equipo.
- **Desafíos:** Pequeña competición entre los alumnos, el o los ganadores obtendrán un premio.
- **Regalos:** Premios que se le da a los alumnos una vez conseguidos sus objetivos.

Por otro lado, las técnicas dinámicas están más enfocadas a la motivación del propio alumno para jugar y seguir a delante. Algunas de las más utilizadas son las siguientes:

- **Logro:** Superación o satisfacción personal.
- **Recompensas:** Obtener un beneficio merecido.
- **Competición:** Por el simple afán de competir y ser mejor que los demás.

Por último, el concepto de Gamificación no es el de crear un juego como tal, sino aprovecharnos de los diferentes sistemas de puntuación-objetivos-recompensas-clasificación que normalmente componen a los mismos.

## 2.2 Herramientas de uso en la educación

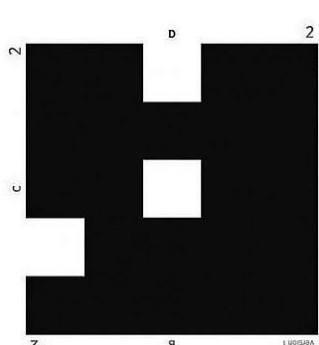
Después ver en que consiste la Gamificación, he hecho una búsqueda de aplicaciones/herramientas que se basan en ello y hacen uso de cuestionarios. A continuación, veremos unos cuantos ejemplos:

### 2.2.1 Plickers

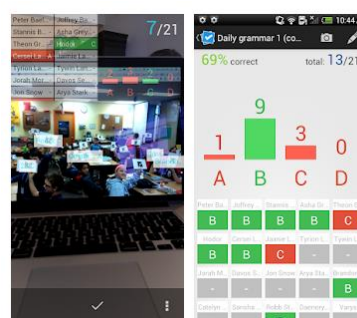
La primera de las herramientas que vamos a conocer es Plickers (**figura 2.1**), la cual es una herramienta de realidad aumentada que permite realizar cuestionarios y evaluaciones de una manera lúdica sin la necesidad de excesivos dispositivos tecnológicos, ya que con un solo dispositivo por aula sería suficiente.



**Figura 2.1** Plickers



**Figura 2.2** Plickers 2



**Figura 2.3** Plickers 3

¿Cómo funciona esta aplicación?

En esta App existen dos roles, el de encuestado (alumno) y el de encuestador (profesor). El caso de encuestador, como su propio nombre indica, será la persona que lanzará la pregunta a toda la clase. Mientras que los encuestados son las personas que responde haciendo uso de la tarjeta impresa que tenemos en la **figura 2.2**, orientando la ficha según la opción escogida. En este tipo de cuestionario no se tiene miedo a que se copien, ya que todas las formas son diferentes para cada uno de los alumnos. Una vez todos los alumnos tienen la ficha levantada con la respuesta escogida, con un dispositivo móvil, se procederá a escanear las tarjetas (**figura 2.3**). Una vez escaneadas, se podrán ver las soluciones al instante y todos los alumnos podrán ver sus resultados.

### 2.2.2 Quizlet

Otra de las herramientas que hace uso de los cuestionarios es Quizlet (**figura 2.4**), que es una herramienta de estudio tanto individual como grupal, que permite a los profesores crear un conjunto de tarjetas con contenidos educativos, incluyendo texto e imágenes, las cuales serán compartidas con alumnos mediante un seguido de actividades/juegos (**figura 2.5**).



Figura 2.4 Quizlet

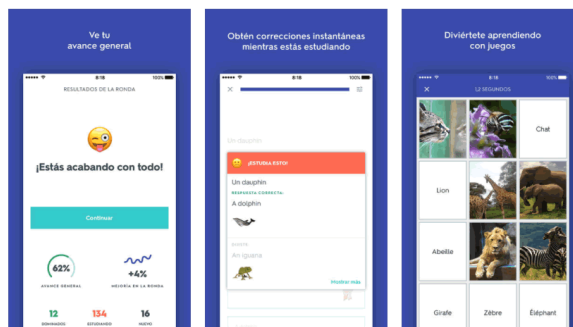


Figura 2.5 Quizlet 2

Esta herramienta tiene diferentes dinámicas para evaluar el aprendizaje de los alumnos (flashcards, scatter y speller), pero analizando a fondo la más interesante es la de test.

Con la actividad de test, a los alumnos les aparecerá un examen con todos los conceptos que hayan trabajado y al final les dará una evaluación completa. En el examen se pueden crear preguntas abiertas, de emparejar, de opción múltiple/falso. Por último, esta herramienta es también una buena opción para crear exámenes escritos.

### 2.2.3 Flipquiz

La siguiente herramienta que vamos a conocer es Flipquiz (**figura 2.6**), la cual es una aplicación web que permite crear tablas de preguntas con una puntuación habitual del 100 al 500 (podría ser superior) y de esta forma crear un juego interactivo con los alumnos usando un único ordenador. En la aplicación tenemos dos tipos de juego:

- **Flashcard:** Recorre pregunta a pregunta y una vez tengan la respuesta a la pregunta se le da la vuelta y se puede ver la respuesta correcta
- **Presentación:** Se colocarán todas las preguntas en una tabla (sin ver que preguntas son, solo se verán las puntuaciones) y los alumnos se tienen que separar por equipos. En el momento en el que el profesor seleccione una de las preguntas, los alumnos tendrán un tiempo para contestarla y finalmente el profesor repartirá una puntuación entre los equipos en función a sus respuestas.



Figura 2.6 Flipquiz

### 2.2.4 Socrative

La última herramienta de la que vamos a hablar es Socrative (**figura 2.7**), la cual es una aplicación diseñada para crear preguntas con sus correspondientes respuestas a tiempo real o asíncrono. Esta app es similar a Classpip, ya que tiene la versión de profesor y la de alumno. Se pueden crear cuestionarios individuales, grupales, concursos de preguntas y hasta gincanas en tiempo real.

Por otro lado, esta aplicación ofrece la modalidad “Space Race” (**figura 2.8**), la cual permite que los estudiantes compitan entre ellos mediante lo que ellos denominan carreras virtuales, con una nave asignada. El profesor formulara un conjunto de preguntas y los estudiantes de forma individual o grupal, irán avanzando con la nave en función de las respuestas acertadas y del tiempo que hayan tardado en responder.

Por último, esta app permite descargar los resultados del alumno en diferentes formatos (Excel, pdf, enviando al correo, ...).



**Figura 2.7** Socrative



**Figura 2.8** Socrative 2

Para acabar este apartado, cabe mencionar una de las herramientas más conocidas, Kahoot. No la he añadido en este apartado ya que trabajos anteriores de otros alumnos (ej: David Hernández López) ya han hablado de ella y he preferido hablar de otras aplicaciones menos conocidas. Pero esto no quiere decir que luego a la hora de pensar ideas para mi módulo no me haya inspirado en la metodología de trabajo de esta herramienta.

## 2.3 Lluvia de ideas

Las diferentes ideas que me han surgido después de hacer un análisis en profundidad de las diferentes herramientas que existen, de realizar un estudio en profundidad sobre el funcionamiento de Classpip y las diferentes funcionalidades que tiene por el momento, son las siguientes:

- Un juego de cuestionarios que durase toda una semana. El cual trataría de un cuestionario diario y el niño que consiguiera más puntos a lo largo de la semana obtendría más puntos para la asignatura.
- La posibilidad de que el profesor cree el cuestionario con las características que vea más convenientes para la ocasión, como la

cantidad de preguntas que desee, con el nivel de dificultad oportuno, la ordenación de las preguntas y respuestas (si cada alumno tiene las preguntas y las respuestas con un diferente orden). De esta forma haríamos un cuestionario completamente adaptativo.

- Un cuestionario modo Kahoot, en el cual todos los alumnos contesten a la vez y la puntuación vaya en función a la velocidad de respuesta y acierto de cada uno de los alumnos. Y la clasificación final del juego se traduciría en puntos para la clasificación del juego de competición que estuvieran realizando.
- La posibilidad de que en los test puedan corregir los fallos que hayan hecho (con una penalización en la nota, para no poder conseguir la máxima puntuación), para que de esta forma aprendan de sus errores y puedan solucionarlos.
- Un tipo de cuestionario por equipos, el cual con que los conteste una persona del equipo, sería suficiente para puntuar todos.
- La opción de hacer cuestionarios de una única asignatura o que las preguntas puedan llegar a ser de diferentes asignaturas y por lo tanto puedan aplicar todos los conocimientos que tengan.
- A la hora de realizar un examen, hay muchas formas de hacerlo. Como por ejemplo haciendo pregunta a pregunta y se va guardando/corrigiendo automáticamente una vez contestada, y pasamos a la siguiente. Por otro lado, tenemos la posibilidad de tener todas las preguntas en una misma página y una vez contestadas todas les damos a finalizar el cuestionario y nos sale automáticamente nuestra nota.

Después de esta lluvia de ideas, vamos a ver que es Classpip y después nos centraremos en el módulo de cuestionarios.



## CAPÍTULO 3. INTRODUCCIÓN A CLASSPIP

En este capítulo vamos a ver en qué consiste Classpip. Inicialmente, veremos la descripción del proyecto y seguidamente su arquitectura actual.

### 3.1 Descripción del proyecto Classpip

Como hemos visto anteriormente, Classpip es un proyecto que fue creado por el Departamento de Arquitectura de Computadores de la Universidad Politécnica de Cataluña (UPC), la arquitectura de la aplicación fue desarrollada en un TFM (Trabajo final de master) por un alumno en 2016. A día de hoy, está formado por cuatro proyectos (Dashboard, Services, Mobile-Profesor y Mobile-Alumno), que veremos en detalle más adelante, y estos contienen un amplio abanico de módulos implementados dentro, cada uno de ellos desarrollados por diferentes alumnos en diferentes proyectos (TFG). Cada módulo, ha sido creado por separado, pero con el mismo fin, de darle valor a Classpip y acabar relacionando cada uno de ellos entre sí para dar el máximo de posibilidades de uso. Como hemos visto en la introducción, hay dos roles, el de profesor y el del alumno, y cada uno dispondrá de un seguido de funcionalidades dentro de cada uno de ellos.

Classpip es una herramienta creada con el fin de introducir la gamificación dentro de las aulas. Como hemos visto en el capítulo anterior, la gamificación consiste en la introducción de las mecánicas propias de los juegos en escenarios que no son juegos, para intentar motivar a los alumnos a realizar cosas que tal vez no tengan ganas de hacer. Actualmente Classpip permite gestionar los siguientes tipos de juego:

#### 3.1.1 Los juegos de Classpip

Los 3 tipos de juegos que Classpip tiene operativos actualmente son los siguientes:

- Juego de puntos: El profesor asignará puntos a los alumnos en función de sus méritos (ayuda a un compañero, buen comportamiento, puntualidad, asistencia, ...) o aleatoriamente, como podemos ver en la **figura 3.1**. Estos puntos se irán acumulando dentro de un ranking y quien consiga una mayor puntuación obtendrá un privilegio (poder escuchar música mientras realizan ejercicios en clase, sentarse en el sitio que desee, ...).

## Pantalla para asignar Puntos

Seleccione  
 Tipo de punto  
 Puntualidad

Número de puntos:

<input type="checkbox"/>	Posición	Nombre	Primer Apellido	Segundo Apellido	Puntos	Nivel
<input type="checkbox"/>	1	Pol	Peinado	Alcalde	0	
<input type="checkbox"/>	2	Sergio	Sánchez	Plaza	0	
<input type="checkbox"/>	3	Albert	Lillo	Méndez	0	
<input type="checkbox"/>	4	Noa	García	Esteve	0	
<input type="checkbox"/>	5	Berta	Junqué	Llaudet	0	
<input type="checkbox"/>	6	Victor	Pérez	Batlle	0	
<input type="checkbox"/>	7	David	Balboa	Mato	0	

**Figura 3.1** Asignación de puntos

- **Juego de colección:** Este tipo de juego consiste en conseguir completar una colección de cromos, de las que previamente habrá creado el profesor dentro del módulo de colecciones. El profesor irá repartiendo cromos de la colección establecida en función de los méritos de los alumnos o aleatoriamente (como vemos en la **figura 3.2**). Los alumnos, por su parte, tratarán de completar la colección y podrán intercambiar entre ellos.

## Asignar Cromos

Asignación manual    Asignación aleatoria    Asignación según puntos

Número de cromos a asignar:

**Figura 3.2** Asignación de cromos

- **Juego de competición:** La finalidad del juego, como su propio nombre indica, es la competir entre los diferentes alumnos o equipos que estén inscritos dentro de un grupo. La competición irá jornada a jornada, y los resultados dependerán de las normas que estipule el profesor (como vemos en la **figura 3.3**), ya que el ganador de una jornada puede ser escogido manualmente, aleatoriamente o incluso en función de los puntos que lleve dentro de un juego de puntos paralelo. Finalmente ganará quien tenga más punto y conseguirá el premio que decida el profesor. Por último, dentro de este juego tenemos diferentes modalidades: Liga, Fórmula Uno y Torneo.

## Pantalla para asignar el Ganador

Test2: Juego De Competición Liga Individual

Seleccione  
Jornada  
1

Modo  
Aleatoriamente

Asignar ganadores aleatoriamente

Alumno 1	Alumno 2	Empate
<input type="checkbox"/> Sergio Sánchez Plaza	<input type="checkbox"/> David Balboa Mato	<input type="checkbox"/> Empate
<input type="checkbox"/> Albert Lillo Méndez	<input type="checkbox"/> Victor Pérez Battle	<input type="checkbox"/> Empate
<input type="checkbox"/> Noa Garcia Esteve	<input type="checkbox"/> Berta Junqué Llaudet	<input type="checkbox"/> Empate

**Figura 3.3** Asignación ganador juego de competición

Por otro lado, todos los tipos de juegos pueden ser disputados tanto individualmente como por equipos. Todo dependerá del objetivo que tenga el profesor pensado para el juego. Todos los participantes interaccionaran a través de la web, o bien desde un ordenador (por ejemplo, el profesor asignando puntuaciones) o desde el dispositivo móvil (por ejemplo, los alumnos consultando la puntuación que llevan por el momento). Por último, para gestionar estos juegos el profesor dispone de los recursos que veremos a continuación.

### 3.1.2 Colecciones

El profesor en este recurso tendrá la opción de crear, editar y eliminar colecciones. Estas son básicamente un conjunto de cromos que agrupamos dentro de un álbum según las características de cada uno de ellos, las cuales estipula el profesor. Posteriormente a crear una colección, esta puede ser asignada dentro de un juego de colección y el profesor ira repartiendo los cromos (especificando el que quiere asignar o uno al azar) a los alumnos que desee, ya sea por méritos propios o porque les toca aleatoriamente.

### 3.1.3 Puntos e Insignias

En este recurso el profesor podrá crear, modificar y eliminar puntos e insignias. Y posteriormente tendrá opción de asignarlas a todos los alumnos o equipos que desee. Estos puntos e insignias se pueden utilizar como premio dentro de los juegos que tenga un grupo en función de la posición o puntuación de cada uno de los alumnos o equipo.

### 3.1.4 Alumnos, Grupos y Equipos

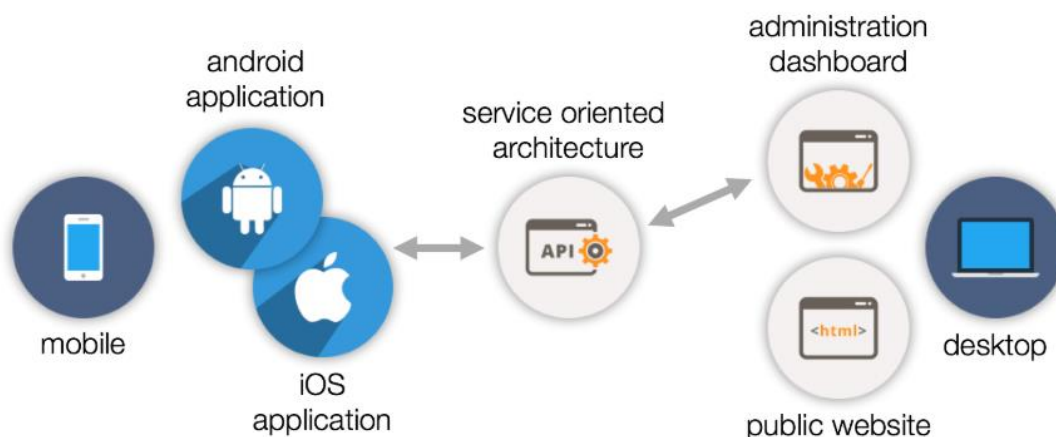
Esta es una de las partes esenciales, ya que sin alumnos no tendríamos opción a la realización de ninguno de los juegos.

En cuanto al módulo encargado de la gestión de alumnos. El profesor tiene libre elección para agregar o eliminarlos. En el caso de que desee agregar tiene dos opciones: La primera es introducir los alumnos uno a uno. Y la segunda de una forma masiva, con la cual podría añadir una lista de alumnos de una sentada. Una vez tenemos un buen conjunto de alumnos se puede proceder a crear un grupo, lo cual es otra parte importante, ya que cada uno de los juegos se establece por grupo.

Para empezar, el profesor creará los grupos que desee agrupándolos por asignaturas (por ejemplo: grupo de Música de 2º ESO, Ingles de 4º ESO, ...) y añadiendo el número de alumnos que vea conveniente. Una vez tenga los grupos creados, el profesor tiene completa libertad para hacer cualquier modificación dentro de cada uno de los grupos (editar o eliminar) y por otro lado tendrá la opción de crear equipos de alumnos en un mismo grupo. Para acabar, dentro de un grupo el profesor puede crear y asignar juegos. Estos juegos los podrá crear individuales o por equipos y se disputarán según lo que especifique el profesor.

## 3.2 Arquitectura del proyecto

Desde que se empezó a trabajar en Classpip, siempre ha tenido la misma arquitectura (**figure 3.4**), o mejor dicho estaba enfocado a dicha arquitectura.



**Figura 3.4** Arquitectura de Classpip

Actualmente Classpip está formado por 4 proyectos, los cuales vamos a ver a continuación:

- ❖ Classpip-services: Este es nuestro punto de partida, una API REST conectada con la base de datos, la cual ofrece al resto de proyectos los

servicios de acceso a datos (por ejemplo, obtener alumnos de un grupo o las preguntas de un profesor). Esta API ha sido creada mediante LoopBack, que es un framework open-source para Node.js que nos permite crear aplicaciones de forma sencilla. Esta tecnología corre en una dirección html a la cual accederán nuestros proyectos (Dashboard y Mobile) para crear, modificar, solicitar y eliminar información.

- ❖ Classpip-dashboard: Esta parte está centrada en la aplicación web (WebApp). Está basada en Node, Express, Angular y Angular material para la interfaz del usuario. Este repositorio está enfocado para el profesor, ya que con la ayuda de una WebApp le será más sencillo crear y establecer todos los cambios que crea necesarios en los módulos del proyecto (por ejemplo, añadir alumnos, crear colecciones o crear cualquier tipo de juego que esté disponible).

Los dos proyectos siguientes, son los que hacen referencia a la aplicación móvil tanto del profesor como del alumno. Esta aplicación, está hecha con Ionic, que es un framework para el desarrollo de aplicaciones móviles híbridas, mezclado con otras herramientas como Angular y Angular Material para facilitar el desarrollo de la interfaz de usuario. Por otro lado, Ionic hace uso de Apache Cordova, el cual nos proporciona acceso a las herramientas de los dispositivos, como la cámara, las funcionalidades del lector de la huella dactilar, el GPS, ...

- ❖ Mobile-Profesor: Este proyecto, como su propio nombre indica, está enfocado a la aplicación móvil del profesor. Con la ayuda de esta app el profesor tiene la opción de realizar algunas funciones desde un dispositivo móvil (por ejemplo, mirar la clasificación del juego de liga de los alumnos, asignar puntos o cromos).
- ❖ Mobile-Alumno: Este proyecto, como vemos está enfocado al alumno. Es la app con la cual el alumno tendrá posibilidad de interactuar con los diferentes juegos que tenga disponibles (por ejemplo, consultar su clasificación, intercambiar cromos o responder a un cuestionario de un juego de cuestionario).

Una vez vistos todos los proyectos que forman parte de Classpip, para desarrollar el módulo de cuestionarios tendré que hacer modificaciones en Services añadiendo todo lo necesario para la base de datos, en el Dashboard para la creación de cuestionarios y del juego en sí, y por último en el móvil del alumno para que pueda realizar el juego de cuestionario. Todas estas modificaciones podremos verlas en los siguientes capítulos.

## CAPÍTULO 4. OBJETIVOS Y PLAN DE TRABAJO

En este apartado vamos a ver los objetivos planteados y el plan de trabajo sobre los mismos. Para ver cómo está el trabajo organizado y que plan de actuación tendré.

### 4.1 Objetivos

Analizando Classpip, tiene un conjunto de módulos funcionado que le aportan un gran valor como aplicación y que son útiles entre sí. El trabajo que tengo planteado es el de desarrollar un módulo de cuestionario, el cual considero que es uno de los más importantes, ya que te puede dar infinidad de funcionalidades y se puede incorporar en diversos de los módulos que hay operativos por el momento y en los que están por implementar, como por ejemplo se puede incorporar dentro de un juego de competición tipo liga para definir el resultado final de una jornada según la nota que haya obtenido cada uno de los alumnos. El módulo se puede dividir en 3 bloques: el de preguntas en el cual nos permitirá crearlas con las características que se especifiquen, el de cuestionario que incorporará las preguntas creadas previamente y las características propias del cuestionario, y para acabar el módulo de juego de cuestionario que usará un cuestionario y establecerá las normas del juego, como por ejemplo la puntuación. Cada uno de estos módulos por separado otorga valor a la aplicación, como, por ejemplo, el de preguntas que puede ser utilizado dentro de un geocaching o para hacer un pequeño kahoot. Solo de imaginarlo se pueden apreciar la cantidad de posibilidades que hay dentro de ellos.

Por otro lado, otra parte importante es la de intentar facilitar el trabajo a futuros desarrolladores, ya que si el código está bien documentado y hay videos explicando cómo funciona, será mucho más sencillo añadir nuevas funcionalidades o módulos al proyecto.

A continuación, voy a enumerar los objetivos que tengo del trabajo ordenadamente para después poder realizar un plan de trabajo más específico.

1. Estudio de herramientas que hagan uso de cuestionarios.
2. Establecer con el tutor funcionalidades del módulo.
3. Realización de los primeros mockups y diseño de la base de datos del proyecto.
4. Creación y despliegue del módulo de preguntas en Dashboard.
5. Creación y despliegue del módulo de cuestionario en Dashboard.
6. Creación y despliegue del módulo de juego de cuestionario en Dashboard.
7. Creación y despliegue del móvil del alumno.
8. Finalización primer prototipo. El cual constará del despliegue completo del módulo de juego de cuestionario tanto en el dispositivo del alumno como en Classpip.
9. Añadido de las siguientes funcionalidades en el móvil del alumno:
  - a. Retrospectiva del cuestionario completo del alumno.

- b. Guardar la respuesta de cada una de las preguntas del cuestionario.
10. Finalización prototipo 2.
11. Diferentes pruebas de la aplicación.

## 4.2 Plan de trabajo

Ahora veremos el plan del trabajo del proyecto, en el cual como podremos observar en la **figura 4.1** está estructurado en objetivos por semana (hay que tener en cuenta que entre la semana 6 y 7 están los exámenes de medio cuatrimestre junto con la semana santa y hay un margen más amplio de tiempo). Este plan de trabajo me sirve como hoja de ruta para tener en un único lugar todas mis metas y tareas que debo realizar, junto con el tiempo estipulado para cada una de ellas. Esto me ayudara a administrar mejor mis recursos y a que el proceso sea más eficiente. (La semana uno hace referencia a la semana del 17 de febrero y la semana 15 al 8 de junio)

	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	S13	S14	S15
1															
2															
3															
4															
5															
6															
7															
8															
9															
10															
11															

**Figura 4.1** Plan de trabajo

## CAPÍTULO 5. DISEÑO FUNCIONAL

En este capítulo vamos a ver en que consiste el módulo de cuestionario, primero veremos qué es un juego de cuestionario, seguidamente la mecánica que tendrá el juego junto con la explicación de cada una de sus partes, y para acabar veremos qué funcionalidades añadiré dentro de los diferentes proyectos.

### 5.1 Módulo de cuestionarios

El objetivo principal de este módulo es el de utilizar el concepto de cuestionario dándole un sentido y un enfoque más educativo y competitivo. Estará incorporado dentro de Classpip con total disponibilidad para cualquier uso que desee el maestro, de tal manera que el profesor pueda hacer uso de ello en el momento y la forma que le parezca más adecuada.

Un juego de cuestionario es aquel en el cual un grupo de alumnos compiten por sacar la mejor nota y el que gana se lleva un premio o recompensa, según lo que estipule el profesor. Analizando este concepto de juego, he visto que hay infinidad de posibilidades para la hora de realizarlo (como hemos visto en la lluvia de ideas que tenemos en el apartado 2.3) y después de analizar cada una de estas posibilidades y de hablar con los tutores, hemos decidido desarrollar lo siguiente:

#### 5.1.1 Mecánica del juego

Previamente a poder jugar, el profesor tendrá que realizar un seguido de acciones para poder crearlo y dejarlo todo listo. Primero creará cada una de las preguntas que quiera incorporar dentro del cuestionario, las cuales serán de tipo respuesta múltiple (1 respuesta correcta y 3 incorrectas), seguidamente creará el cuestionario y añadirá todas las preguntas que desee, y para finalizar creará el juego añadiendo el cuestionario generado anteriormente y las normas del juego que quiera establecer. Una vez el juego este creado y activado, el alumno podrá jugarlo desde su dispositivo móvil de forma individual y en el momento que lo finalice recibirá la nota obtenida junto con un pequeño feedback de cada una de las respuestas que haya puesto.

En los siguientes puntos vamos a ver más en detalle que características tendrá cada una de las partes del juego de cuestionario.

#### 5.1.2 Las preguntas

Este es el punto de partida de un buen juego de cuestionario ya que sin preguntas no sería posible realizar un cuestionario. A la hora de crear la pregunta el profesor tendrá que definir los siguientes campos:

- ❖ Título (breve información de la pregunta, ej: Tema 1 de Física)
- ❖ Pregunta



- ❖ Temática (Hace referencia a la asignatura)
- ❖ Respuesta correcta
- ❖ 3 respuestas incorrectas
- ❖ Feedback respuesta correcta
- ❖ Feedback respuesta incorrecta

Como podemos ver tenemos dos campos de feedback, estos campos son para cuando el alumno finalice el examen, tendrá una retrospectiva de las preguntas que ha contestado y de esta forma saber si están bien o si están mal, el porqué de estarlo.

Por último, al crear cada una de las preguntas se irán guardando en “Mis Preguntas”, y es donde accederemos para después poder hacer uso de ellas.

### 5.1.3 Los cuestionarios

Después de haber creado/añadido el número de preguntas oportunas, el profesor procederá a crear el cuestionario en el cual especificará lo siguiente:

- ❖ Título
- ❖ Descripción
- ❖ El número de preguntas que decida (no hay ni máximo ni mínimo número de preguntas).

Cada profesor tendrá un conjunto de preguntas creadas, con sus correspondientes respuestas, y guardadas en “Mis Preguntas” (como he explicado anteriormente), y es ahí donde irá a buscar las preguntas que desee añadir al cuestionario que esté generando.

Por otro lado, cada uno de los cuestionarios que cree los tendrá guardados en “Mis Cuestionarios”, para después poderles dar uso en el juego de cuestionario.

### 5.1.4 El juego de cuestionario

Este es el último paso que nos queda por dar, y es posiblemente el más importante ya que aquí el profesor especificará las normas que tiene el juego de cuestionario y su funcionamiento.

A la hora de crear un juego de cuestionario, tendrá que especificar lo siguiente:

- ❖ Cuestionario a usar
- ❖ Puntuación
- ❖ Presentación

Cuando escoja el cuestionario que desee utilizar, lo irá a buscar a “Mis Cuestionarios”, que como hemos visto anteriormente, es el sitio donde están todos los cuestionarios guardados.

Cuando establezca la puntuación tendrá que definir lo siguiente: por un lado, tendrá que establecer la puntuación que quiere darle a cada una de las preguntas. Y por otro, la penalización que quiera asignarle a cada pregunta errónea (en el caso de no querer penalizar, poniendo un 0 no descontaría puntos).

La presentación (u ordenación), hace referencia a la ordenación con la cual le aparezca a los alumnos las preguntas y las respuestas. Habrá tres posibilidades: Todo con el mismo orden, preguntas desordenadas o preguntas y respuestas desordenadas.

Por último, el juego de cuestionario una vez creado entrará en modo preparado, y cuando el profesor quiera hacer uso de él solo tendrá que activarlo y cuando pase el tiempo que crea conveniente finalizarlo, en ese momento pasara a juego inactivo.

## **5.2 Funciones a realizar desde el Dashboard**

Ahora que hemos visto en que va a consistir el módulo vamos a ver de qué funcionalidades dispondrá el profesor dentro del Dashboard. Para empezar, hay que decir que va a ser el centro de operaciones, ya que aquí va a ser donde crearemos todas y cada una de las partes del cuestionario y del juego.

El profesor tendrá opción de crear, editar y eliminar tanto preguntas, como cuestionarios, como juegos de cuestionario. Para ello, creará un componente para crear preguntas y otro para crear cuestionarios, así como los correspondientes componentes de “Mis Preguntas” y “Mis Cuestionarios” para poder ver, editar y eliminar, dando así completa libertad de edición al profesor. Por otro lado, el profesor tendrá opción de crear el juego de cuestionario con las características que hemos visto previamente. Una vez creado, y antes de activarlo, podrá modificar las características básicas como las puntuaciones o la presentación. Finalmente, una vez activado en el momento en el cual todos los alumnos hayan realizado el cuestionario o cuando se acabe el tiempo, el profesor tendrá opción a finalizarlo y que nadie más pueda jugarlo.

## **5.3 Funciones a realizar desde Mobile-Alumno**

Después de ver en que va a consistir el módulo de cuestionario, y las diferentes funcionalidades que tendrá el Dashboard, es hora de ver que se va a incluir en Mobile-Alumno.

El móvil del alumno va a tener un papel fundamental dentro del módulo de preguntas, ya que, sin el dispositivo no tendría sentido el trabajo, debido a que la intención es la de que cada uno de los alumnos responda el cuestionario de una forma individual e íntima. Dicho esto, vamos a ver las diferentes funcionalidades:

- ❖ Cuando el profesor active el juego de cuestionario, para que los alumnos lo puedan responder, les aparecerá dentro de juegos activos y podrán entrar en él. Una vez empiecen, la primera pantalla que les aparecerá es la de la descripción del cuestionario, las siguientes serán el conjunto de preguntas que haya establecido el profesor (una por página). Irán contestando estas preguntas de una en una, y en el caso de que quieran retroceder a la anterior lo podrán hacer con total libertad. En el momento en el que finalicen el cuestionario recibirán el feedback con la nota y con la retrospectiva de cada una de las preguntas que han contestado, para que puedan saber dónde han acertado y donde han fallado.

## CAPÍTULO 6. DISEÑO DE PANTALLAS

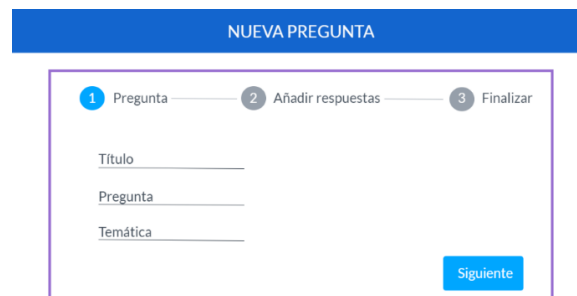
Antes de empezar a desarrollar la aplicación es necesario la creación de unos diseños web, los cuales veremos en este apartado, para ver la dirección que va a tomar el proyecto. Con la ayuda de *marvelapp.com* y *mockflow.com* he podido crear los diseños y a partir de ellos empezaré a trabajar.

A continuación, hay una pequeña explicación del funcionamiento de cada uno de los diseños de los diferentes componentes:

### 6.1 Mockups Dashboard

En la **figura 6.1** podemos ver como empezar a crear una nueva pregunta. Primero estableceremos el título, la pregunta en sí y la temática. En la siguiente página (**figura 6.2**) añadiremos la respuesta correcta, las incorrectas y las correspondientes retrospectivas a cada una de ellas.

Finalmente, si estamos de acuerdo con lo que hemos generado, le daremos al botón de finalizar (**figura 6.3**), el cual mandra un http.post, para guardar la pregunta en nuestra base de datos.



**Figura 6.1** Nueva pregunta



**Figura 6.2** Nueva pregunta 2

**Figura 6.3** Nueva pregunta

En la **figura 6.4** podemos ver una lista de preguntas “Mis Preguntas”. Esta lista esta rellena mediante un `http.get` al servidor y nos muestra todas las preguntas que han sido creadas por el profesor. Como se puede ver, podremos hacer una búsqueda en función del título o de la temática, y en el caso de que deseemos editar o eliminar, lo podríamos hacer desde esta misma página. En el caso que queramos editar, clicaríamos en el botón verde e iríamos a la imagen siguiente (**figura 6.5**), en la que podemos retocar todos y cada uno de los campos de la pregunta, de tal forma que una vez la tengamos lista la guardamos con un `http.put` y quedaría modificada.

Titulo	Temática		
¿Que año empezó la primera gerra mundial?	Historia		
Si un vehiculo va a 20km/h y tiene que recorrer ...	Física		
¿Cual es el numero atomico del oxigeno?	Química		
¿Que animal deja sus huevos enterrados enla playa?	Biología		
¿Que es la superposicion?	Física Cuantica		
¿En que rango de frecuencia emite la FM?	FC		

**Figura 6.4** Mis preguntas

**Figura 6.5** Editar pregunta

En la **figura 6.6** podemos ver cómo crear un cuestionario nuevo. Primero tendríamos que añadir el título y la descripción del cuestionario. Seguidamente (como vemos en la **figura 6.7**) añadiríamos las preguntas que creamos oportunas

al cuestionario (se abrirá una ventana emergente similar a la **figura 6.4**). Y, por último, finalizamos la creación mediante un `http.post` clicando en el botón “Finalizar” (**figura 6.8**).

Figura 6.6 muestra una ventana emergente titulada "NUEVO CUESTIONARIO". En la parte superior, hay una barra de progreso con tres pasos: 1 Descripción (activo), 2 Añadir preguntas, y 3 Finalizar. Debajo, hay campos de texto para "Título" y "Descripción". En la esquina inferior derecha, hay un botón azul que dice "Siguiete".

**Figura 6.6** Nuevo cuestionario

Figura 6.7 muestra la segunda ventana emergente titulada "NUEVO CUESTIONARIO". La barra de progreso ahora muestra el paso 2 "Añadir preguntas" como activo. Debajo, hay un botón naranja que dice "Añadir preguntas" con un icono de lista. En la esquina inferior derecha, hay dos botones: "Anterior" (rojo) y "Siguiete" (azul).

**Figura 6.7** Nuevo cuestionario 2

Figura 6.8 muestra la tercera ventana emergente titulada "NUEVO CUESTIONARIO". La barra de progreso ahora muestra el paso 3 "Finalizar" como activo. Debajo, hay un mensaje que dice "Cuestionario creado correctamente" y "Haga click en 'Finalizar' para guardar los cambios". En la esquina inferior derecha, hay dos botones: "Anterior" (rojo) y "Finalizar" (verde).

**Figura 6.8** Nuevo cuestionario 3

En la **figura 6.9** podemos ver una lista de cuestionarios “Mis Cuestionarios”. Es la lista de todos los cuestionarios generado por el profesor. Se rellena mediante un `http.get` y nos da la opción de entrar a editar el cuestionario que deseemos o incluso eliminarlo.


MIS CUESTIONARIOS		
Título	Descripción	
Primer test	Matematicas	 
Vamos a ver que sabeis	Fisica	 
¿Sereis capaces de aprobar?	Historia	 
Cuanto sabeis de....	Musica	 
El último cuestionario	Aleatorio	 

Figura 6.9 Mis cuestionarios

En las siguientes figuras podremos ver cómo crear un juego de cuestionario. En la **figura 6.10** tendremos que escoger el cuestionario que queremos de la lista de “Mis Cuestionarios”. En la siguiente (**figura 6.11**) añadiremos la puntuación que deseamos dar a cada una de las respuestas correctas y la puntuación que queremos para las incorrectas. Después en la **figura 6.12** estableceremos el orden que queremos para las preguntas y sus correspondientes respuestas. Para acabar, como podemos ver en la **figura 6.13** le daremos a finalizar para guardar nuestro juego de cuestionario, usando un http.post.

JUEGO DE CUESTIONARIO

1 Cuestionario — 2 Puntuación — 3 Presentación — 4 Finalizar

Añadir cuestionario 

Siguiente

Figura 6.10 Juego de cuestionario

JUEGO DE CUESTIONARIO

 Cuestionario — 2 Puntuación — 3 Presentación — 4 Finalizar

Puntuación por pregunta acertada

Penalización por pregunta errónea

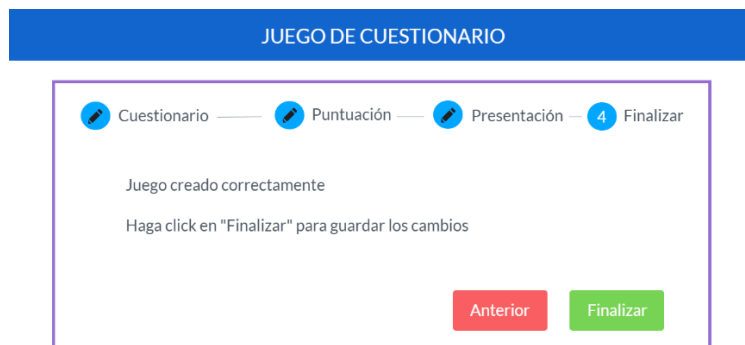
En el caso de que no quiera descontar ningún punto ponga un 0

Anterior Siguiente

Figura 6.11 Juego de cuestionario 2



**Figura 6.12** Juego de cuestionario 3



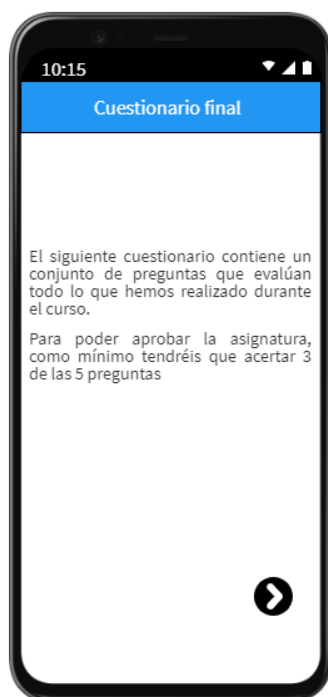
**Figura 6.13** Juego de cuestionario 4

## 6.2 Mockups Mobile-Alumno

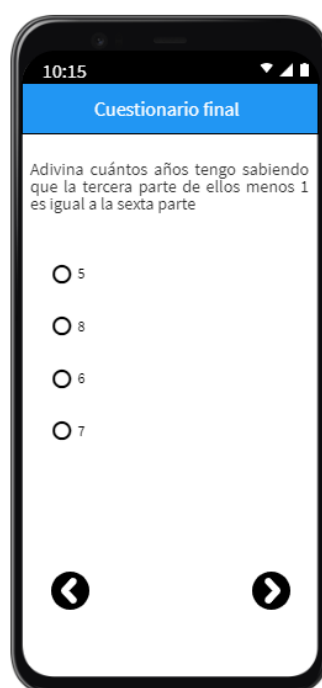
En las siguientes figuras podremos ver lo que aparecerá en la pantalla del alumno a la hora de realizar un cuestionario. Cabe decir que estas pantallas formaran parte del proyecto del móvil del alumno.

La **figura 6.14** nos muestra la descripción del cuestionario que toque realizar y nos da pie a la **figura 6.15** que es la primera pregunta del cuestionario. Una vez contestadas todas las preguntas y pudiendo moverse libremente por cada una de ellas sin límite de veces, llegarían a la **figura 6.16** la cual si clics en “finalizar” acabarían el examen enviando las respuestas con un http.post. Por último, les saldría la **figura 6.17** en la cual podrán ver cómo les ha ido el examen y las respuestas que han acertado/fallado, con su correspondiente feedback.

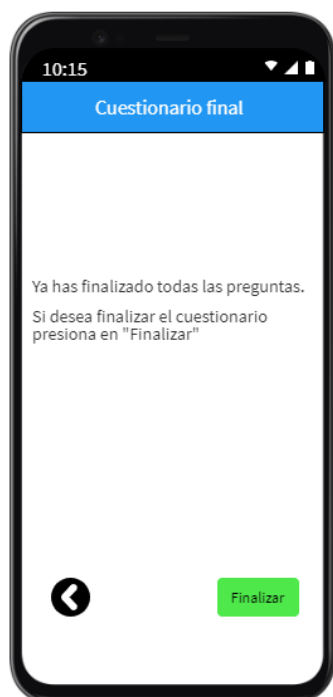




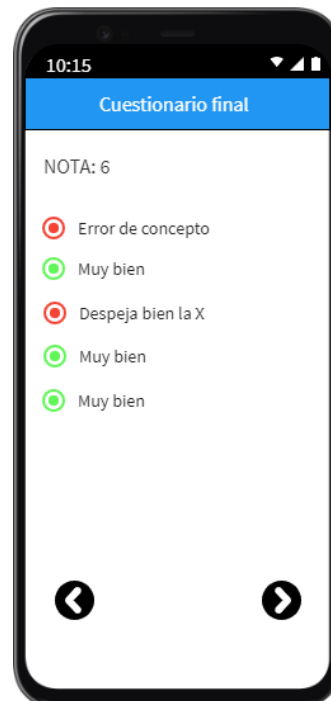
**Figura 6.14** Realización de un cuestionario



**Figura 6.15** Realización de un cuestionario 2



**Figura 6.16** Realización de un cuestionario 3



**Figura 6.17** Resultado del cuestionario

En las siguientes figuras podremos ver los la información de los cuestionarios que ya están finalizados. En la **figura 6.18** podemos ver la lista de los

cuestionarios que han sido finalizados (mediante un `http.get`). Si seleccionamos uno de ellos iríamos a la **figura 6.19** en la cual vemos cual ha sido el resultado de cada uno de los alumnos ordenados en un “ranking” de mejor a peor nota de cada uno de ellos.

Cuestionarios finalizados			
	Título	Aprobados	Suspendidos
1	Primer test	5	2
2	Vamos a ver que sabeis	6	1
3	¿Sereis capaces de aprobar?	3	4
4	Cuanto sabeis de...	4	3
5	Cuestionario final	7	0

**Figura 6.18** Cuestionarios finalizados

Cuestionario final			
Ranking del cuestionario:			
	Name	Nota	Tiempo
1	Ruben Cobos	10	25min
2	David Perez	8	35min
3	Marc Belmonte	6	27min
4	Xavi Fernandez	6	32min
5	Esther Exposito	6	40min

**Figura 6.19** Ranking del cuestionario

### 6.3 Estilos gráficos

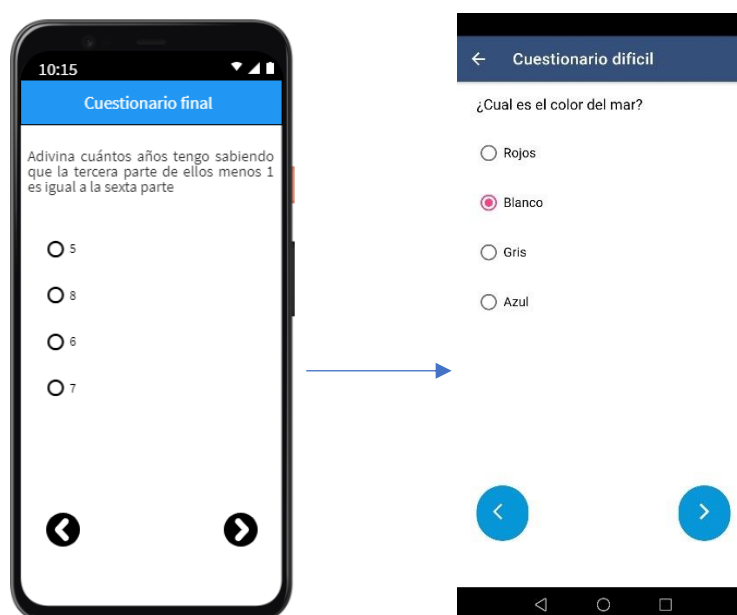
Los estilos gráficos, como su propio nombre indica, es todo aquello que nos otorga información visual y, por lo tanto, es una parte esencial dentro de la interacción con el usuario. Cada uno de los proyectos (Dashboard y Mobile-Alumno) tienen sus propios estilos gráficos (como veremos en el Anexo 2) y cada una de sus pantallas hace uso de ello para que todo vaya hacia la misma dirección.

Cabe decir que gracias a que cuando he diseñado los mockups me he intentado adaptar a los estilos que había visto en los proyectos anteriormente, a la hora de

implementarlo en el Dashboard he conseguido que fuera prácticamente igual a la idea inicial y únicamente he añadido algunos retoques estéticos siguiendo los estilos de cada uno de los proyectos, como podemos ver en la **figura 6.20**. Por otro lado, en el caso del móvil del alumno me ha pasado prácticamente lo mismo, y lo podemos apreciar en la **figura 6.21**.



**Figura 6.20** Transición de mockup a resultado final de “Mis Preguntas”



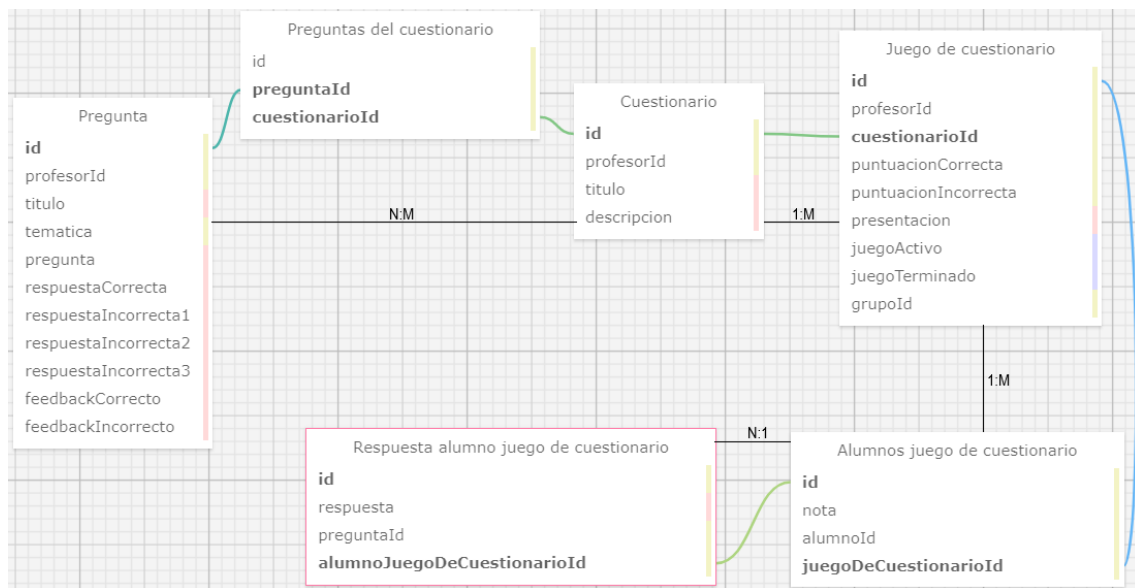
**Figura 6.21** Transición de mockup a resultado final en la realización de un cuestionario

## CAPÍTULO 7. MODELO DE DATOS E IMPLEMENTACIÓN

Este es uno de los capítulos más importantes, ya que aquí veremos cómo organizamos y guardamos los datos dentro del módulo de cuestionario. Primero veremos cómo es la estructura de los datos con los que vamos a trabajar y posteriormente su implementación en el proyecto, junto con algunos ejemplos.

### 7.1 Modelo de datos

Para empezar, vamos a definir bien los modelos de la base de datos y sus correspondientes relaciones. Lo cual, queda resumido en la siguiente **figura 7.1**:



**Figura 7.1** Modelo de datos

En la imagen podemos ver todos los campos que tendrán nuestros modelos de datos, los cuales están relacionados de diferentes formas (X:X) unos con otros. Está hecho de la forma más concisa y sencilla posible, de tal manera que aplicarlo sea relativamente fácil y muy intuitivo a la hora de entender cómo funciona el módulo de cuestionarios.

A continuación, veremos una pequeña explicación de cada uno de los modelos.

#### Pregunta:

Este es el primer paso para después poder generar un buen cuestionario. Esta base de datos contiene toda la información relacionada con una Pregunta (titulo, pregunta, respuestas, ...). Por un lado, tiene una relación 1:M con *Profesor*, ya que un profesor puede tener múltiples preguntas, y por otro, tiene una relación N:M con *Cuestionario*, ya que cada pregunta puede estar usada en múltiples cuestionarios y cada cuestionario tiene múltiples preguntas. Para esta relación

es necesario que un modelo de datos haga de enlace entre las dos. Ese modelo es *Preguntas del cuestionario* y es el siguiente del cual vamos a hablar.

### **Preguntas del cuestionario:**

En esta tabla vamos a relacionar cada una de las preguntas con los cuestionarios y viceversa. Como vemos en la **figura 7.1** guardaremos los identificadores tanto de las preguntas como de los cuestionarios, para que de esta manera tengamos constancia de la relación.

### **Cuestionario:**

Este modelo de datos es la base del juego de cuestionario. Aquí tendremos todos los cuestionarios que han sido creados por el profesor y que estarán disponibles para el momento en el que se decida hacer uso de ellos en un juego. Cada uno de ellos estará formado por un título, descripción, y el identificador del profesor que lo haya creado.

Por un lado, este modelo estará relacionado con *Juego de cuestionario* en 1:M ya que un cuestionario puede ser utilizado en múltiples juegos, y, por otro lado, tiene una relación N:M con preguntas, como hemos visto anteriormente.

### **Juego de cuestionario:**

Esta es la base de datos más importante, ya que es donde el juego va a coger forma y consistencia para poder hacer uso de él. Aquí podremos configurar cada uno de los parámetros del juego (Escoger cuestionario, puntos a sumar y restar, y la ordenación de las preguntas y respuestas).

Esta base de datos tendrá una relación 1:M con *Cuestionario*, ya que cada juego de cuestionario solo puede contener un cuestionario asociado, con *Profesor*, ya que cada profesor puede haber creado múltiples juegos de cuestionario, y con *Grupo* porque puede tener más de un juego de cuestionario un único grupo. También, tendrá una relación N:M con *Alumno*, ya que un alumno puede tener múltiples juegos de cuestionario, y un juego de cuestionario puede tener más de un alumno inscrito. Para hacer constancia de esta relación he creado la siguiente tabla puente que se llama *Alumnos juego de cuestionario* y veremos cómo funciona a continuación.

### **Alumno juego de cuestionario:**

En esta base de datos guardaremos todos los resultados de los cuestionarios (las notas que vayan obteniendo). Podremos buscar el resultado concreto de un alumno en el juego de cuestionario que deseemos.

Esta base de datos hará de puente entre *Juego de cuestionario* y *Alumno*, ya que tendrán una relación N:M, como hemos explicado antes.

Y como podemos ver, tendrá una relación 1:M con *Juego de cuestionario* debido a que cada uno de los resultados del alumno solo pueden estar relacionados con un único juego de cuestionario.

### Respuesta alumno juego de cuestionario:

Para finalizar, tenemos la base de datos en la cual guardaremos todas y cada una de las respuestas de los alumnos en cada una de las preguntas del cuestionario. Esta base de datos tiene una relación 1:M con *Alumno juego de cuestionario*, ya que guardaremos múltiples respuestas y cada una de estas respuestas solo puede hacer referencia a un alumno del juego de cuestionario, y con *Pregunta*, ya que cada una de las preguntas habrá sido contestada múltiples veces.

## 7.2 Implementación modelo de datos

La base de datos ha sido realizada en el repositorio de Classpip-services, que como hemos explicado anteriormente, es nuestra API REST. Este es el repositorio encargado de guardar todos los datos de la aplicación y por lo tanto donde se harán todas las peticiones para que la aplicación funcione a la perfección. Como, por ejemplo, para hacer un login, crear preguntas, los cuestionarios, etc.

Gracias a que la API ha sido generada con LoopBack, la creación de modelos y de relaciones ha sido relativamente fácil, ya que con dos simples comandos se ha podido generar todo y sin ninguna complicación. Dichos comandos son los siguientes:

- *Lb model*: Este comando nos da la opción de generar nuestros modelos, especificando en cada uno de ellos sus diferentes propiedades y el origen de datos al que conectar el modelo. En nuestro caso es una base de datos en memoria.
- *Lb relation*: Este comando nos ayuda a crear las relaciones entre los diferentes modelos que tenemos, ya sean relaciones 1:M o N:M. En el caso de que tengamos una relación 1:M solo tenemos que establecer una relación *hasMany* dentro del modelo que necesitamos. Por otro lado, al crear una N:M tenemos que hacer dos relaciones de *hasMany* entre los modelos principales, y dos relaciones más de tipo *belongsTo* dentro de la tabla de relación.

Para continuar, vamos a ver los modelos creados y las tablas de relación:

### 7.2.1 Preguntas

En esta base de datos guardamos toda la información relacionada con una pregunta y sus relaciones.

```

"properties": {
  "Titulo": { "type": "string", "required": true },
  "Pregunta": { "type": "string", "required": true },
  "Tematica": { "type": "string", "required": true },
  "RespuestaCorrecta": { "type": "string", "required": true },
  "RespuestaIncorrecta1": { "type": "string", "required": true },
  "RespuestaIncorrecta2": { "type": "string", "required": true },
  "RespuestaIncorrecta3": { "type": "string", "required": true },
  "FeedbackCorrecto": { "type": "string", "required": true },
  "FeedbackIncorrecto": { "type": "string", "required": true }
},
"validations": [],
"relations": {
  "cuestionarios": [
    {
      "type": "hasMany",
      "model": "Cuestionario",
      "foreignKey": "preguntaId",
      "options": {
        "nestRemoting": true
      },
      "through": "PreguntaDelCuestionario"
    }
  ],
  "respuestaAlumnoJuegoDeCuestionarios": {
    "type": "hasMany",
    "model": "RespuestaAlumnoJuegoDeCuestionario",
    "foreignKey": "preguntaId"
  }
}

```

**Figura 7.2** Modelo pregunta

Como podemos ver en la **figura 7.2** tiene una relación con *Cuestionario*, que veremos más tarde reflejada en la tabla de *Pregunta del cuestionario*, y otra relación con *Respuesta alumno juego de cuestionario* en la cual, como veremos más tarde, guardaremos las respuestas que han hecho de la pregunta en cuestión.

### 7.2.2 Cuestionario

El siguiente modelo guarda toda la información de un cuestionario (**figura 7.3**), excepto algunos campos, como por ejemplo la puntuación, que se guardaran en el modelo de *Juego de cuestionario*, como hemos visto anteriormente.

```

"properties": {
  "Titulo": { "type": "string", "required": true },
  "Descripcion": { "type": "string", "required": true }
},
"validations": [],
"relations": {
  "Preguntas": {
    "type": "hasMany",
    "model": "Pregunta",
    "foreignKey": "cuestionarioId",
    "options": {
      "nestRemoting": true
    },
    "through": "PreguntaDelCuestionario"
  },
  "juegosDeCuestionario": {
    "type": "hasMany",
    "model": "JuegoDeCuestionario",
    "foreignKey": "cuestionarioId"
  }
}
},

```

**Figura 7.3** Modelo cuestionario

Como podemos apreciar, en este modelo también hay una relación con preguntas que usa la tabla *Pregunta del cuestionario* que veremos a continuación.

### 7.2.3 Pregunta del cuestionario

Este modelo de datos, como he avanzado previamente, recoge todas las relaciones pregunta-cuestionario (como vemos en la **figura 7.4**), lo cual nos ayuda a saber que preguntas están siendo utilizadas dentro de cada uno de los cuestionarios.

```

"properties": {},
"validations": [],
"relations": {
  "Pregunta": {
    "type": "belongsTo",
    "model": "Pregunta",
    "foreignKey": "preguntaId",
    "options": {
      "nestRemoting": true
    }
  },
  "Cuestionario": {
    "type": "belongsTo",
    "model": "Cuestionario",
    "foreignKey": "cuestionarioId",
    "options": {
      "nestRemoting": true
    }
  }
}
},

```

**Figura 7.4** Modelo pregunta del cuestionario

Como podemos ver, no tiene ninguna propiedad, únicamente guarda el identificador de la pregunta con el identificador del cuestionario, para así relacionar unos con otros.



### 7.2.4 Juego de cuestionario

Este modelo contiene todo lo necesario para realizar un buen juego de cuestionario, como podemos ver en la **figura 7.5**.

```
"properties": {
  "NombreJuego": { "type": "string", "required": true },
  "PuntuacionCorrecta": { "type": "number", "required": true },
  "PuntuacionIncorrecta": { "type": "number", "required": true },
  "Presentacion": { "type": "string", "required": true },
  "JuegoActivo": { "type": "boolean", "required": true, "default": false },
  "JuegoTerminado": { "type": "boolean", "required": true, "default": false }
},
"validations": [],
"relations": {
  "alumnos": {
    "type": "hasMany",
    "model": "Alumno",
    "foreignKey": "juegoDeCuestionarioId",
    "options": {
      "nestRemoting": true
    },
    "through": "AlumnoJuegoDeCuestionario"
  }
},
}
```

**Figura 7.5** Modelo juego de cuestionario

Como vemos, este modelo tiene una relación con alumno que queda recogida en una tabla llamada *Alumno juego de cuestionario*, que veremos a continuación.

### 7.2.5 Alumno juego de cuestionario

El siguiente modelo es el de *Alumno juego de cuestionario* (**figura 7.6**) en el cual, como podemos ver, recogemos todos los alumnos que participan en un juego de cuestionario junto con la nota que han sacado en el cuestionario y por otro lado tenemos una relación *hasMany* con *Respuesta alumno juego de cuestionario*, ya que tendrá tantas como preguntas tenga el cuestionario.

Este modelo es muy importante, ya que de aquí se podrán sacar estadísticas en función de los resultados de los alumnos. Por otro lado, tendremos constancia de los alumnos que han jugado y los que aún no lo han hecho.

```

"properties": {
  "Nota": { "type": "number", "required": true }
},
"validations": [],
"relations": {
  "alumno": {
    "type": "belongsTo",
    "model": "Alumno",
    "foreignKey": "alumnoId",
    "options": {
      "nestRemoting": true
    }
  },
  "juegoDeCuestionario": {
    "type": "belongsTo",
    "model": "JuegoDeCuestionario",
    "foreignKey": "juegoDeCuestionarioId",
    "options": {
      "nestRemoting": true
    }
  },
  "respuestaAlumnoJuegoDeCuestionarios": {
    "type": "hasMany",
    "model": "RespuestaAlumnoJuegoDeCuestionario",
    "foreignKey": "alumnoJuegoDeCuestionarioId"
  }
}

```

**Figura 7.6** Modelo alumno juego de cuestionario

### 7.2.6 Respuesta alumno juego de cuestionario

Para finalizar vamos a ver cómo es la estructura del modelo *Respuesta alumno juego de cuestionario*. Como podemos ver en la **figura 7.7** dentro del modelo únicamente tenemos una sola propiedad que es la de la respuesta del alumno y ninguna relación, esto es debido a que las relaciones las hemos establecido dentro de *Pregunta* y *Alumno juego de cuestionario* (como hemos visto previamente) ya que ambas dos tendrán múltiples respuestas.

```

"properties": {
  "Respuesta": {
    "type": "string",
    "required": true
  }
},
"validations": [],
"relations": {},

```

**Figura 7.7** Modelo respuesta alumno juego de cuestionario

## 7.3 Consultas a la base de datos

Una vez implementados todos los modelos y las relaciones de cada uno de ellos, procedemos a arrancar la API. Loopback tiene implementado un cliente API Explorer que nos da opción a ir probando el correcto funcionamiento de cada uno de los modelos y en el caso de algún fallo lo podemos visualizar y corregir. Por otro lado, es un buen sitio para ir comprobando como van los datos y para poder obtener las URL necesarias para poder realizar cualquier tipo de petición desde nuestro frontend.

Los diferentes tipos de consultas son los siguientes:

- **GET:** Petición que únicamente con la URL nos proporciona los datos que solicitemos.
- **POST:** Para esta petición necesitamos la URL y el objeto a guardar, esto nos permitirá añadir datos.
- **PUT:** Con la URL, un identificador y el objeto, nos permite modificar los valores de un objeto en concreto.
- **DELETE:** Con la URL y el identificador, nos permite eliminar de la base de datos la información que especifiquemos.

Seguidamente, veremos un ejemplo de cada una de las consultas, para poder entender mejor su funcionamiento.

### 7.3.1 Ejemplo POST

Para hacer un POST en la API Explorer únicamente necesitamos el objeto el cual queremos añadir y la URL sobre la cual lo añadiremos a la base de datos. Si vemos la **figura 7.8** vemos un ejemplo de POST y nos retorna un 200 lo que quiere decir que todo ha ido bien y que se ha guardado correctamente.

The screenshot shows the API Explorer interface for a POST request to the endpoint `/Preguntas`. The interface is divided into several sections:

- Parameters:** A table with columns: Parameter, Value, Description, Parameter Type, and Data Type. The 'data' parameter is shown with a JSON body:
 

```
{
  "Titulo": "test",
  "Pregunta": "test",
  "Tematica": "test",
  "RespuestaCorrecta": "test",
  "RespuestaIncorrecta1": "test",
  "RespuestaIncorrecta2": "test",
  "RespuestaIncorrecta3": "test",
  "FeedbackCorrecto": "test",
  "FeedbackIncorrecto": "test",
  "id": 12,
  "profesorId": 1
}
```

 The 'Parameter content type' is set to `application/json`.
- Request URL:** The URL is `http://localhost:3000/api/Preguntas`.
- Response Body:** The response is a JSON object:
 

```
{
  "Titulo": "test",
  "Pregunta": "test",
  "Tematica": "test",
  "RespuestaCorrecta": "test",
  "RespuestaIncorrecta1": "test",
  "RespuestaIncorrecta2": "test",
  "RespuestaIncorrecta3": "test",
  "FeedbackCorrecto": "test",
  "FeedbackIncorrecto": "test",
  "id": 12,
  "profesorId": 1
}
```
- Response Code:** The response code is `200`.

**Figura 7.8** Ejemplo petición POST

### 7.3.2 Ejemplo GET

A la hora de hacer una petición GET desde nuestra API Explorer tenemos diferentes opciones: la primera es la de solicitar toda la información de un mismo modelo de datos, y para llevarla a cabo únicamente necesitamos la URL. Y la segunda, es una petición de un objeto único, la cual se llevará a cabo introduciendo su identificador (como podemos ver en la **figura 7.9**).

GET /Preguntas/{id} Find a model instance by {{id}} from the data source.

Parameter	Value	Description	Parameter Type	Data Type
id	12	Model id	path	string
filter		Filter defining fields and include - must be a JSON-encoded string ({"something":"value"})	query	string

Try it out! [Hide Response](#)

**Curl**

```
curl -X GET --header 'Accept: application/json' 'http://localhost:3000/api/Preguntas/12'
```

**Request URL**

```
http://localhost:3000/api/Preguntas/12
```

**Response Body**

```
{
  "Titulo": "test",
  "Pregunta": "test",
  "Tematica": "test",
  "RespuestaCorrecta": "test",
  "RespuestaIncorrecta1": "test",
  "RespuestaIncorrecta2": "test",
  "RespuestaIncorrecta3": "test",
  "FeedbackCorrecto": "test",
  "FeedbackIncorrecto": "test",
  "id": 12,
  "profesorId": 1
}
```

**Response Code**

```
200
```

**Figura 7.9** Ejemplo petición GET

### 7.3.3 Ejemplo PUT

A la hora de hacer un PUT desde la API Explorer tenemos diferentes opciones: añadir, reemplazar o modificar. En el caso de la **figura 7.10** vemos como se modifica una pregunta ya existente, operación para la que necesitamos los datos de la pregunta, ya sean modificados o no, el identificador y la URL.

**PUT** /Preguntas/{id} Replace attributes for a model instance and persist it into the data source.

Parameter	Value	Description	Parameter Type	Data Type
id	12	Model id	path	string

**Parameters**

**data**

```
{
  "Titulo": "test modificado",
  "Pregunta": "test",
  "Tematica": "test",
  "RespuestaCorrecta": "test",
  "RespuestaIncorrecta1": "test",
  "RespuestaIncorrecta2": "test",
  "RespuestaIncorrecta3": "test",
  "FeedbackCorrecto": "test",
  "FeedbackIncorrecto": "test",
  "id": 12,
  "profesorId": 1
}
```

Parameter content type: **application/json**

**Model instance data**

**body**

**Model**

**Example Value**

```
{
  "Titulo": "string",
  "Pregunta": "string",
  "Tematica": "string",
  "RespuestaCorrecta": "string",
  "RespuestaIncorrecta1": "string",
  "RespuestaIncorrecta2": "string",
  "RespuestaIncorrecta3": "string",
  "FeedbackCorrecto": "string",
  "FeedbackIncorrecto": "string",
  "id": 0,
  "profesorId": 0
}
```

[Try it out!](#) [Hide Response](#)

**Request URL**

http://localhost:3000/api/Preguntas/12

**Response Body**

```
{
  "Titulo": "test modificado",
  "Pregunta": "test",
  "Tematica": "test",
  "RespuestaCorrecta": "test",
  "RespuestaIncorrecta1": "test",
  "RespuestaIncorrecta2": "test",
  "RespuestaIncorrecta3": "test",
  "FeedbackCorrecto": "test",
  "FeedbackIncorrecto": "test",
  "id": 12,
  "profesorId": 1
}
```

**Response Code**

200

Figura 7.10 Ejemplo petición PUT

### 7.3.4 Ejemplo DELETE

Como último ejemplo de petición tenemos el DELETE, el cual para hacerlo desde la API Explorer tenemos que introducir el identificador del objeto que queremos eliminar y mediante la URL se elimina (**figura 7.11**).

**DELETE** /Preguntas/{id} Delete a model instance by {{id}} from the data source.

**Parameters**

Parameter	Value	Description	Parameter Type	Data Type
id	12	Model id	path	string

[Try it out!](#) [Hide Response](#)

**Curl**

```
curl -X DELETE --header 'Accept: application/json' 'http://localhost:3000/api/Preguntas/12'
```

**Request URL**

```
http://localhost:3000/api/Preguntas/12
```

**Response Body**

```
{
  "count": 1
}
```

**Response Code**

```
200
```

**Figura 7.11** Ejemplo petición DELETE

## CAPÍTULO 8. IMPLEMENTACIÓN APLICACIONES

En este apartado vamos a ver cómo han sido el proceso de desarrollo del proyecto en cada uno de los repositorios. Para empezar, veremos paso a paso como se ha desplegado el módulo en el Dashboard y en el dispositivo móvil del alumno. Seguidamente, veremos un punto con la cuantificación del trabajo realizado, es decir, veremos un resumen del número de componentes creados o cuantas líneas de código he escrito. Por último, tenemos una explicación del algoritmo más complejo de este proyecto para facilitar su entendimiento para futuras mejoras.

### 8.1 Dashboard y móvil del alumno.

Ahora vamos a ver los diferentes tipos de ficheros que he creado tanto en el Dashboard como en el dispositivo móvil del alumno, los cuales están gestionados en el repositorio Classpip-dashboard (WebApp) y en Mobile-Alumno (Ionic App) como hemos visto en apartados anteriores. Para el desarrollo del módulo he seguido el siguiente orden: creación de modelos, después del componente y para acabar los servicios necesarios de cada uno. Y vamos a verlos a continuación:

#### 8.1.1 Modelos

Este es el paso inicial que he seguido a la hora de empezar a desarrollar en cualquiera de los proyectos. He creado los modelos en función a los parámetros que he estipulado dentro de Classpip-services para que en el momento en el que lance una petición siga la misma estructura de datos y no nos salte un error. Para crearlos, he seguido la estructura definida de Classpip y los he situado dentro de la carpeta donde están localizados el resto de modelos (como podemos ver algunos en la **figura 8.1** y **figura 8.2**).

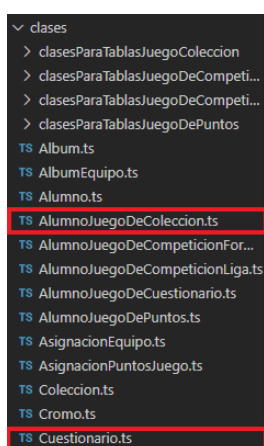


Figura 8.1 Modelos Dashboard

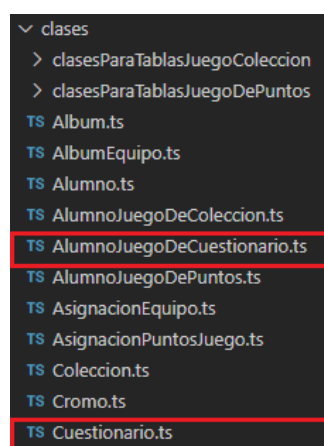


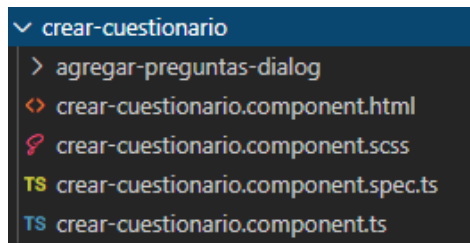
Figura 8.2 Modelos móvil del alumno

Si seleccionamos uno de ellos podremos ver las características del modelo de datos y, por lo tanto, cuáles son los campos necesarios para lo hora de crearlo.

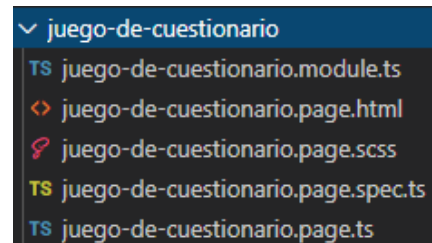
Una vez bien generados todos los modelos, pasamos a crear los componentes.

### 8.1.2 Componentes

El siguiente paso a seguir, es la creación de los componentes (en el caso de Ionic se llaman páginas, pero es el mismo concepto), pero este paso lo vamos a separar en dos: vista y controlador. El primero es la edición del html junto con el css de cada uno de los componentes, que haría referencia a la vista, y una vez lo tenemos funcionando pasamos a editar el controlador para aplicar todas las funcionalidades que creamos necesarias al componente. Podemos ver un ejemplo en la **figura 8.3** (Dashboard) y **figura 8.4** (móvil alumno).



**Figura 8.3** Componente crear-cuestionario



**Figura 8.4** Pagina juego de cuestionario

Si nos centramos más en las funcionalidades, tenemos dos tipos (**figura 8.5**): Primero, las que se ejecutan en el momento en el cual cargamos el componente, que sería las que situamos dentro del *ngOnInit()*, y se usan para cargar toda la información que será necesaria dentro de nuestro componente, como por ejemplo cargar todas las preguntas de un cuestionario, las normas que tendrá el juego, rescatar el identificador del profesor,... Y por otro lado tenemos las funciones que se ejecutan dependiendo de cómo interactuemos con el html, como por ejemplo si queremos crear una pregunta al clicar en el botón de crear se ejecutaría la función *CrearPregunta()* y recogerá todos los valores estipulados dentro del código html para así poder generar bien la pregunta con las características que hayamos escrito dentro del formulario.



```

ngOnInit() {
    this.profesorId = this.sesion.DameProfesor().id;

    // tslint:disable-next-line:no-string-literal
    this.URLVueltaInicio = this.route.snapshot.queryParams['URLVueltaInicio'] || '/inicio';

    this.myForm = this._formBuilder.group({
        tituloPregunta: ['', Validators.required],
        preguntaPregunta: ['', Validators.required],
        tematicaPregunta: ['', Validators.required]
    });

    this.myForm2 = this._formBuilder.group({
        respuestaCorrecta: ['', Validators.required],
        respuestaIncorrecta1: ['', Validators.required],
        respuestaIncorrecta2: ['', Validators.required],
        respuestaIncorrecta3: ['', Validators.required],
        feedbackCorrecto: ['', Validators.required],
        feedbackIncorrecto: ['', Validators.required]
    });

    //CREAMOS Y GUARDAMOS LA PREGUNTA CON LOS VALORES AÑADIDOS
    CrearPregunta() {
        let tituloPregunta: string;
        let preguntaPregunta: string;
        let tematicaPregunta: string;
        let respuestaCorrecta: string;
        let respuestaIncorrecta1: string;
    }
}

```

**Figura 8.5** Controlador componente Dashboard

Para poder realizar correctamente una función de guardar, coger, cambiar o eliminar (POST, GET, PUT, DELETE), es necesario un servicio el cual hará una petición a la base de datos en funciona lo que necesitemos. Y es lo que voy a explicar en el apartado siguiente.

### 8.1.3 Servicios

Por último, tenemos los servicios. Los cuales nos proporcionan acceso a la información de la base de datos desde el Dashboard o móvil del alumno. Dentro de servicios, tenemos 3 clases de servicios:

- Primero, tenemos los servicios de sesión (**figura 8.6**), en los cuales como su propio nombre indica, guardamos la información de la sesión, como el identificador del profesor, el juego seleccionado, la lista de preguntas... Y la podemos recoger y utilizar en el momento que sea necesario.

```

public DamePregunta(): Pregunta {
    return this.pregunta;
}

public TomaPregunta(pregunta: Pregunta) {
    this.pregunta = pregunta;
}

public DameListaCuestionarios(): any {
    return this.listaCuestionarios;
}

public TomaListaCuestionarios(listaCuestionarios: any){
    this.listaCuestionarios = listaCuestionarios;
}

```

**Figura 8.6** Sesión service

- Segundo tenemos los servicios de cálculos, los cuales se ejecutan en el momento en el cual tenemos que realizar una operación más compleja, como por ejemplo en el momento en el que queremos eliminar un cuestionario también tenemos que eliminar las relaciones de preguntas que están guardadas dentro de ese cuestionario (**figura 8.7**).

```
public EliminarCuestionario(): any {
  const eliminaObservable = new Observable ( obs => {
    this.peticionesAPI.BorraCuestionario(
      this.sesion.DameProfesor().id,
      this.sesion.DameCuestionario().id)
    .subscribe() => {
      this.EliminarPreguntasDelCuestionario();
      obs.next ();
    });
  });
  return eliminaObservable;
}

// ESTA FUNCIÓN RECUPERA TODAS LAS PREGUNTAS DEL CUESTIONARIO QUE VAMOS A BORRAR Y DESPUÉS LAS BORRA. ESTO LO HACEMOS PARA NO
// DEJAR RELACIONES PREGUNTADEL CUESTIONARIO QUE NO NOS SIRVEN EN LA BASE DE DATOS
private EliminarPreguntasDelCuestionario() {
  // Pido las preguntasDelCuestionario correspondientes al cuestionario que voy a borrar
  this.peticionesAPI.DamePreguntasDelCuestionarioCuestionario(this.sesion.DameCuestionario().id)
  .subscribe( preguntasDelCuestionario => {
    if (preguntasDelCuestionario[0] !== undefined) {
      // Una vez recibo las preguntas del cuestionario, las voy borrando una a una
      for (let i = 0; i < preguntasDelCuestionario.length; i++) {
        this.peticionesAPI.BorraPreguntaDelCuestionario(preguntasDelCuestionario[i].id)
        .subscribe() => {
          console.log('Pregunta del cuestionario borrada correctamente');
        });
      }
    } else {
      console.log('no hay preguntas en el cuestionario');
    }
  });
}
```

**Figura 8.7** Cálculos service

- Por último, tenemos los servicios de peticiones a la API (**figura 8.8**). Aquí es donde hacemos todas y cada una de las peticiones, ya sean GET, POST, PUT o DELETE.

```
//Preguntas
public CreaPregunta(pregunta: Pregunta, profesorId: number): Observable<Pregunta> {
  console.log(pregunta, profesorId);
  return this.http.post<Pregunta>(this.APIUrlProfesores + '/' + profesorId + '/preguntas', pregunta);
}

public DameTodasMisPreguntas(profesorId: number): Observable<Pregunta[]> {
  return this.http.get<Pregunta[]>(this.APIUrlProfesores + '/' + profesorId + '/preguntas');
}

public BorrarPregunta(preguntaId: number): Observable<any> {
  return this.http.delete<any>(this.APIUrlPreguntas + '/' + preguntaId);
}

public ModificaPregunta(pregunta: Pregunta, profesorId: number, preguntaId: number): Observable<Pregunta> {
  return this.http.put<Pregunta>(this.APIUrlProfesores + '/' + profesorId + '/preguntas/' + preguntaId, pregunta);
}

//Cuestionarios
public CreaCuestionario(cuestionario: Cuestionario, profesorId: number): Observable<Cuestionario> {
  console.log('Cuestionario: ' + cuestionario);
  return this.http.post<Cuestionario>(this.APIUrlProfesores + '/' + profesorId + '/cuestionarios', cuestionario);
}

public ModificaCuestionario(cuestionario: Cuestionario, profesorId: number, cuestionarioId: number): Observable<Cuestionario> {
  return this.http.put<Cuestionario>(this.APIUrlProfesores + '/' + profesorId + '/cuestionarios/' + cuestionarioId, cuestionario);
}

public DameTodosMisCuestionarios(profesorId: number): Observable<Cuestionario[]> {
  return this.http.get<Cuestionario[]>(this.APIUrlProfesores + '/' + profesorId + '/cuestionarios');
}

public BorraCuestionario(profesorId: number, cuestionarioId: number): Observable<any> {
  return this.http.delete<any>(this.APIUrlProfesores + '/' + profesorId + '/cuestionarios/' + cuestionarioId);
}
```

**Figura 8.8** Peticiones API service

## 8.2 Cuantificación trabajo realizado

Este punto intenta mostrar una visión más concreta de la carga de trabajo que ha supuesto la implementación del módulo, ya que para una persona externa al código es complicado cuantificar el trabajo realizado. Cabe decir que si nos centramos en los números de la **figura 8.9**, tal vez, parece que no sea mucho, pero sí que ha supuesto una alta inversión de tiempo, con tal de que todo funcione correctamente.

En la siguiente tabla tenemos una serie de datos, los cuales algunos de ellos no son completamente exactos pero lo suficiente para entender la alta carga de faena que ha supuesto este trabajo. Cabe decir que dentro de esta tabla no está cuantificado el trabajo realizado dentro de Classpip-services ya que no se puede cuantificar de la misma manera.

	Dashboard	Móvil del alumno	Total
Clases	6	5	11
Componentes	13	1	14
Funciones implementadas	>100	>25	>125
Líneas de código modificadas	>4300	>1700	>6000

**Figura 8.9** Cuantificación trabajo realizado

## 8.3 Algoritmo crítico

En este apartado veremos en detalle el algoritmo implementado más complejo, el cual merece una explicación más detallada por si futuros desarrolladores necesitan ayuda, puedan usar esto como referencia.

El algoritmo en cuestión es el de ordenación de preguntas y respuestas de los cuestionarios y vamos a proceder a ver cómo funciona.

### 8.3.1 Ordenación de preguntas y respuestas

Para poder realizar un juego de cuestionario tenemos dos pasos iniciales, el primero es el de coger el juego de cuestionario y el segundo el de solicitar las preguntas correspondientes a dicho juego.

Para empezar, en el momento en el que tenemos el cuestionario adquirimos el tipo de presentación que tiene estipulada. En este punto, ya podemos solicitar las preguntas. Seguidamente guardaremos estas preguntas en un vector de preguntas. Por otro lado, tenemos un vector predefinido en el cual indica en que posición estará la pregunta correcta en cada una de las preguntas del cuestionario, como podemos ver en la **figura 8.10**.

```
PreguntasCuestionario: Pregunta[];
descripcion: string = '';
puntuacionCorrecta: number;
puntuacionIncorrecta: number;
respuestasPosibles: string[] = [];
RespuestaEscogida: string;
RespuestasAlumno: string[] = [];
Nota: number = 0;
puntuacionMaxima: number = 0;
NotaInicial: string = '';
feedbacks: string[] = [];

//Con este array establecemos la posicion donde
ordenRespuestaCorrecta: number[] = [2, 3, 0,
```

**Figura 8.10** Variables juego de cuestionario

Una vez tenemos esto claro, procedemos a hacer 3 tipos de recorridos en función del tipo de ordenación. Los diferenciaremos en los siguientes apartados:

### 8.3.1.1 Preguntas y respuestas mismo orden

Este es el recorrido más sencillo, ya que no tendremos que hacer ninguna reordenación de ninguno de los campos. Lo único que tenemos que ir modificando es el vector de respuestas para que salgan las correctas al alumno en el momento en el cual proceda a contestar una de las preguntas (en la **figura 8.11** vemos como recoge las primeras respuestas de la pregunta 1 y las carga en respuestas posibles).

```
this.peticionesAPI.DamePreguntasCuestionario(this.juegoSeleccionado.cuestionarioId)
.subscribe(res => {
  if (this.juegoSeleccionado.Presentacion !== 'Mismo orden para todos') {
    this.ordenarRespuestas(res)
  } else {
    this.PreguntasCuestionario = res;
  }
  this.respuestasPosibles.push(res[0].RespuestaIncorrecta1);
  this.respuestasPosibles.push(res[0].RespuestaIncorrecta2);
  this.respuestasPosibles.push(res[0].RespuestaIncorrecta3);
  this.respuestasPosibles.splice(this.ordenRespuestaCorrecta[0], 0, res[0].RespuestaCorrecta);
```

**Figura 8.11** Petición de preguntas y carga de las primeras respuestas posibles

En el momento en el cual el alumno quiere pasar a la siguiente pregunta, clicando en el botón del html que guía hacia la siguiente, se ejecuta *cambioRespuestaSiguiente()* (**figura 8.12**). Esta función hace dos procesos importantes: el primero es el de guardar la respuesta escogida de la pregunta que ha respondido y la guarda dentro de un vector de respuestas, y la segunda es la de buscar las siguientes 4 respuestas posibles de la pregunta siguiente y las substituye dentro del vector de respuestas posibles.

```

cambioRespuestasSiguiente(i: number) {
  this.RespuestasAlumno.splice(i, 1, this.RespuestaEscogida);
  if (this.RespuestasAlumno[i+1] !== undefined) {
    this.RespuestaEscogida = this.RespuestasAlumno[i+1];
  } else if ((i+1) !== this.PreguntasCuestionario.length){
    this.RespuestaEscogida = '';
  } else {
    this.RespuestaEscogida = this.RespuestasAlumno[i];
  }

  if ((i+1) !== this.PreguntasCuestionario.length && this.juegoSeleccionado.Presentacion !== 'Preguntas y respuestas desordenadas'){
    this.respuestasPosibles = [];
    i = i + 1;
    this.respuestasPosibles.push(this.PreguntasCuestionario[i].RespuestaIncorrecta1);
    this.respuestasPosibles.push(this.PreguntasCuestionario[i].RespuestaIncorrecta2);
    this.respuestasPosibles.push(this.PreguntasCuestionario[i].RespuestaIncorrecta3);
    this.respuestasPosibles.splice(this.ordenRespuestaCorrecta[i], 0, this.PreguntasCuestionario[i].RespuestaCorrecta);
  }

  if ((i+1) !== this.PreguntasCuestionario.length && this.juegoSeleccionado.Presentacion === 'Preguntas y respuestas desordenadas') {
    this.numeroDeRespuestas = this.numeroDeRespuestas + 4;
    this.respuestasPosibles = [];
    for (var i = this.numeroDeRespuestas; i < (this.numeroDeRespuestas + 4); i++){
      this.respuestasPosibles.push(this.todasRespuestas[i])
    }
  }
}

```

**Figura 8.12** Función cambio de respuestas

Por último, cabe mencionar que si el alumno quiere regresar a la respuesta anterior, el proceso sería exactamente el mismo, pero cargando las respuestas de la pregunta anterior dentro del vector de respuestas posibles.

### 8.3.1.2 Preguntas desordenadas

Este es el recorrido más complicado de procesar, ya que no es trivial. En el momento en el cual adquirimos las preguntas, tenemos que comprobar el tipo de presentación, y en el caso de que no sea “mismo orden para todos” procedemos a ejecutar la función *ordenarRespuestas()* (como podemos ver en la **figura 8.11**). Lo que hace esta función es lo siguiente (**figura 8.13**):

1. Mezcla todas las preguntas que tiene el cuestionario para que cada alumno le salgan con un orden distinto.
2. Compara el nuevo orden de preguntas con el orden inicial, en el caso de que sea diferente (cosa que es lo esperado), reordena el vector de orden de respuestas correctas con el nuevo orden de las preguntas, de tal forma que en este caso solo las preguntas estén desordenadas y las respuestas de cada pregunta van a estar colocadas en la misma posición para cada alumno. Por ejemplo, si en el orden inicial la pregunta 1 tenía la respuesta correcta en la casilla 3, y ahora la pregunta 1 está situada como pregunta 4, pues después de pasar por este bucle la pregunta 4 tendrá la respuesta correcta situada dentro de la casilla 3.

```
ordenarRespuestas(preguntas: Pregunta[]) {
  this.PreguntasCuestionarioOrdenadas = preguntas;
  this.PreguntasCuestionario = this.desorden Preguntas(this.PreguntasCuestionarioOrdenadas);

  if (this.PreguntasCuestionarioOrdenadas !== this.PreguntasCuestionario ){
    for (var i = 0; i < this.PreguntasCuestionarioOrdenadas.length; i++) {
      for (var j = 0; j < this.PreguntasCuestionario.length; j++) {
        if (this.PreguntasCuestionarioOrdenadas[i] === this.PreguntasCuestionario[j]) {
          this.nuevaOrdenacion.splice(this.nuevaOrdenacion[i], 0, this.ordenRespuestaCorrecta[j]);
        }
      }
    }
    this.ordenRespuestaCorrecta = this.nuevaOrdenacion;
  }
}
```

**Figura 8.13** Función ordenarRespuestas

### 8.3.1.3 Preguntas y respuestas desordenadas

Por último, llegamos a la posibilidad de presentación de todo desordenado, en este caso, lo primero que haremos es desordenar las preguntas, como ha pasado en el caso anterior, y lo segundo desordenar cada bloque de 4 respuestas y añadirlas todas dentro de un vector de respuestas, como podemos ver en la **figura 8.14**. Este vector de respuestas nos ira proporcionando las 4 respuestas posibles de cada pregunta en el momento en el cual el alumno seleccione siguiente/anterior dentro del cuestionario.

```
if (this.juegoSeleccionado.Presentacion === 'Preguntas y respuestas desordenadas') {
  this.respuestasPosibles = this.desordenRespuestas(this.respuestasPosibles);
  this.todasRespuestas = this.respuestasPosibles;

  var i = 1;
  while (i < this.PreguntasCuestionario.length) {
    this.mezclaRespuestas.push(res[i].RespuestaIncorrecta1);
    this.mezclaRespuestas.push(res[i].RespuestaIncorrecta2);
    this.mezclaRespuestas.push(res[i].RespuestaIncorrecta3);
    this.mezclaRespuestas.push(res[i].RespuestaCorrecta);
    this.mezclaRespuestas = this.desordenRespuestas(this.mezclaRespuestas);

    this.todasRespuestas = this.todasRespuestas.concat(this.mezclaRespuestas);

    i++;
    this.mezclaRespuestas = [];
  }
}
```

**Figura 8.14** Mezcla de respuestas

## CAPÍTULO 9. PRUEBAS

Este capítulo hace referencia a uno de los aspectos más importantes del desarrollo de una aplicación, las pruebas. Este es el último paso para comprobar si todo funciona a la perfección o si hay que hacer alguna modificación para que todo sea correcto y que ninguna de las funcionalidades falle a la hora de que un usuario haga uso de la aplicación.

Para poder hacer estas comprobaciones he desarrollado una lista de pruebas, que veremos en el apartado 9.1, que se tienen que ir haciendo paso a paso para ver que todo funciona y con el fin de intentar cubrir todos y cada uno de los posibles aspectos desarrollados o con opción de uso. Por otro lado, estos pasos podrán ser utilizados por futuros desarrolladores de la aplicación para asegurarse de que todo sigue funcionando bien y aparte para conocer cómo funciona y las diferentes utilidades que tiene incorporadas.

Por último, he grabado un video tutorial en el que se puede ver cómo funciona el módulo implementado, y en el cual se prueban todas y cada una de las funcionalidades en el orden que hay estipulado en las pruebas. El video puede encontrarse en el siguiente enlace:

- <https://youtu.be/6bHWOVflxEQ>

### 9.1 Lista de pruebas

En este punto, tenemos la lista de pruebas separadas por bloques. De tal forma que, si un desarrollador solo tiene que probar una de ellas, le será más sencillo.

#### 9.1.1 Pruebas módulo de preguntas

1. Vamos a la pestaña de preguntas del navbar y seleccionamos “Crear Pregunta”.
2. Rellenamos los campos necesarios para crear una pregunta hasta que le demos al botón de finalizar.
3. Navegamos hasta la pestaña de “Mis Preguntas”.
4. Seleccionamos la opción de editar pregunta sobre la que hemos creado.
5. Cambiamos algún campo de la pregunta y guardamos.
6. En el caso de que no tengamos que crear un cuestionario, volvemos a “Mis Preguntas” procedemos a eliminar la pregunta creada.

#### 9.1.2 Pruebas módulo de cuestionario

7. Vamos a “Crear Cuestionario”.
8. Rellenamos los primeros campos de cuestionario y le damos a “siguiente”.
9. Ahora tendremos la opción de añadir preguntas, pero antes de añadir tenemos que intentar navegar a cualquier sitio, para ver si nos salta la

guardia y nos da opción a escoger si queremos abandonar o no (Le damos a “NO”).

10. Seleccionamos “Añadir preguntas”.
11. Añadimos una pregunta y después la quitamos (tal que el cuestionario no tenga ninguna pregunta).
12. Le damos a siguiente y posteriormente finalizamos la creación del cuestionario.
13. Vamos a “Mis Cuestionarios” y clicamos en “editar” el que hemos creado.
14. Una vez dentro primero seleccionamos el botón de “Añadir/Eliminar Preguntas” y una vez aquí añadimos todas las preguntas que queramos al cuestionario.
15. Ahora cambiamos cualquiera de los campos del cuestionario y le damos a “Aceptar Cambios”.
16. En el caso de que no queramos crear un juego de cuestionario, volvemos a “Mis Cuestionarios” y eliminamos el cuestionario creado.

### 9.1.3 Pruebas módulo juego de cuestionario

1. Vamos a “Mis Grupos” y entramos en uno.
2. Vamos a juegos.
3. Cambiamos de “Lista de Juegos” a “Crear Juego”.
4. Ponemos nombre al juego.
5. Escogemos “Juego de Cuestionario”.
6. Rellenamos todos los campos hasta que le demos a “Finalizar”.
7. En la lista de juegos vamos a “Juegos Preparados” y seleccionamos el nuevo.
8. Una vez dentro clicamos en “Información” para ver que todos los campos han sido guardados correctamente.
9. Sin salir de información, hacemos algún cambio y guardamos.
10. Activamos el juego clicando en “Activar”.
11. Vamos al móvil del alumno y seleccionamos el juego de cuestionario que hemos activado.
12. Una vez empecemos intentaremos tirar atrás con el botón superior izquierdo. Nos saltará un aviso de si queremos salir o no y seleccionamos “No”.
13. Acabamos el cuestionario revisando los feedbacks de las respuestas para ver que estén bien puestos, en función si hemos acertado, fallado o no contestado.
14. Una vez volvamos a la página inicial del móvil, intentamos volver a entrar en el juego. Y vemos que no nos deja entrar otra vez, ya que lo hemos hecho antes.
15. Desde el Dashboard, entramos en el juego activo y lo finalizamos.
16. Ahora desde el móvil, el juego nos aparecerá en juegos finalizados y procedemos a ver el ranking de notas.
17. Por último, entramos en el juego finalizado desde el Dashboard y lo borramos.

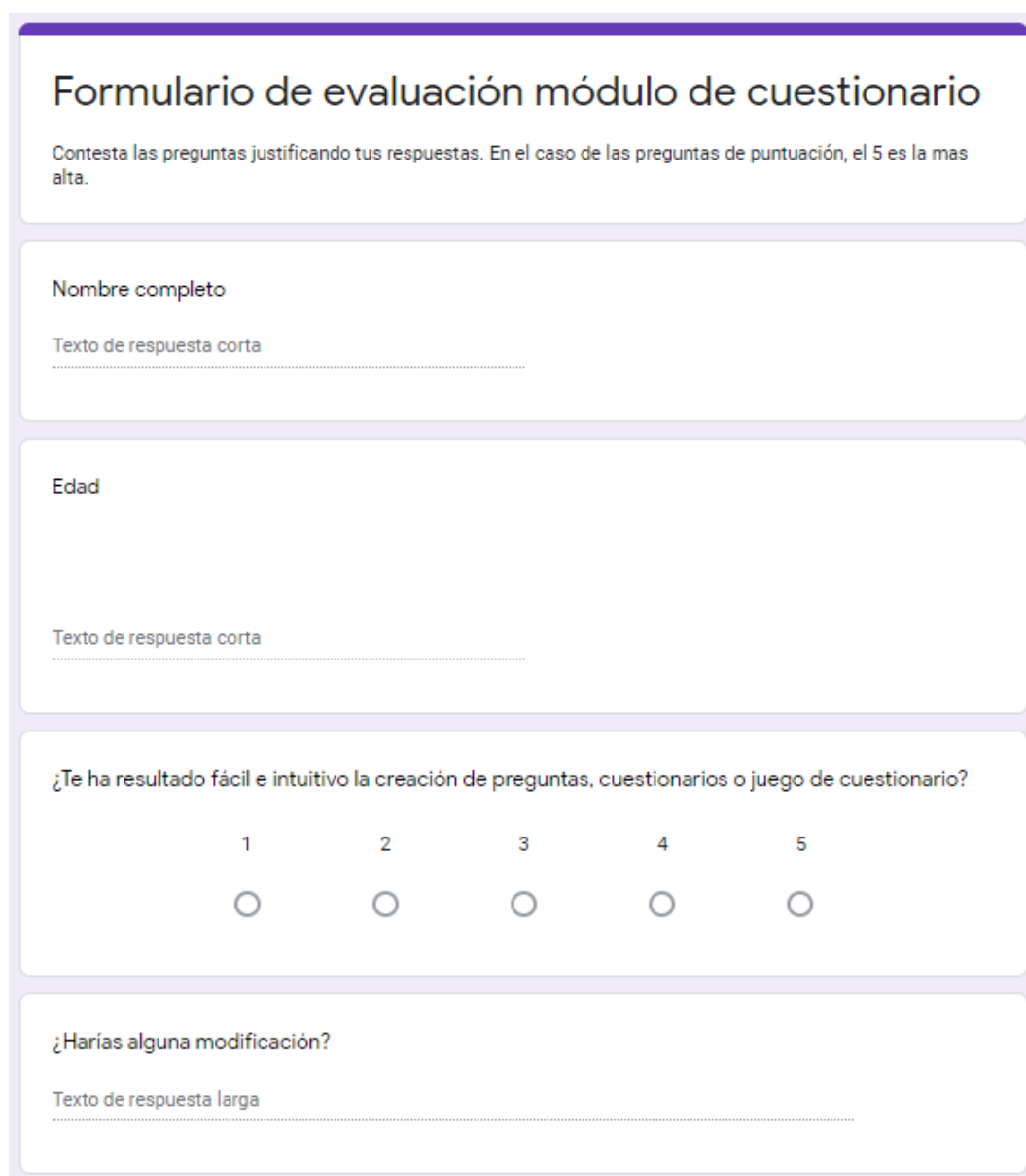


## CAPÍTULO 10. EVALUACIÓN

En este capítulo vamos a evaluar el módulo implementado. Para esta evaluación me va a ayudar tres personas, las cuales van a probar la aplicación y luego les pasare un formulario para que puedan evaluarla. Finalmente veremos una valoración en función de los resultados obtenidos y los comentaremos.

### 10.1 Modelo de evaluación

Para poder hacer una correcta evaluación de la aplicación, he creado el formulario de la **figura 10.1**. Con la ayuda de este formulario y con la correspondiente retrospectiva de las personas que lo contesten podré sacar unas conclusiones más reales sobre el proyecto y su funcionamiento, también surgirán posibles mejoras y retoques que pudiera tener el módulo implementado.



Formulario de evaluación módulo de cuestionario

Contesta las preguntas justificando tus respuestas. En el caso de las preguntas de puntuación, el 5 es la mas alta.

Nombre completo

Texto de respuesta corta

Edad

Texto de respuesta corta

¿Te ha resultado fácil e intuitivo la creación de preguntas, cuestionarios o juego de cuestionario?

1 2 3 4 5

○ ○ ○ ○ ○

¿Harias alguna modificación?

Texto de respuesta larga

...

¿Te ha resultado fácil e intuitivo la modificación de preguntas, cuestionarios o juego de cuestionario?

1 2 3 4 5

☐ ☐ ☐ ☐ ☐

¿Harias alguna modificación?

Texto de respuesta larga

.....

¿El modelo de pregunta te ha parecido correcto? ¿Harias algún cambio?

Texto de respuesta larga

¿Consideras que el modelo de cuestionario o juego de cuestionario contiene todo lo necesario?

Texto de respuesta larga

¿A la hora de realizar el cuestionario desde la aplicación móvil, te ha resultado fácil e intuitivo?

1 2 3 4 5

☐ ☐ ☐ ☐ ☐

¿Harias alguna modificación en la aplicación móvil?

Texto de respuesta larga

.....

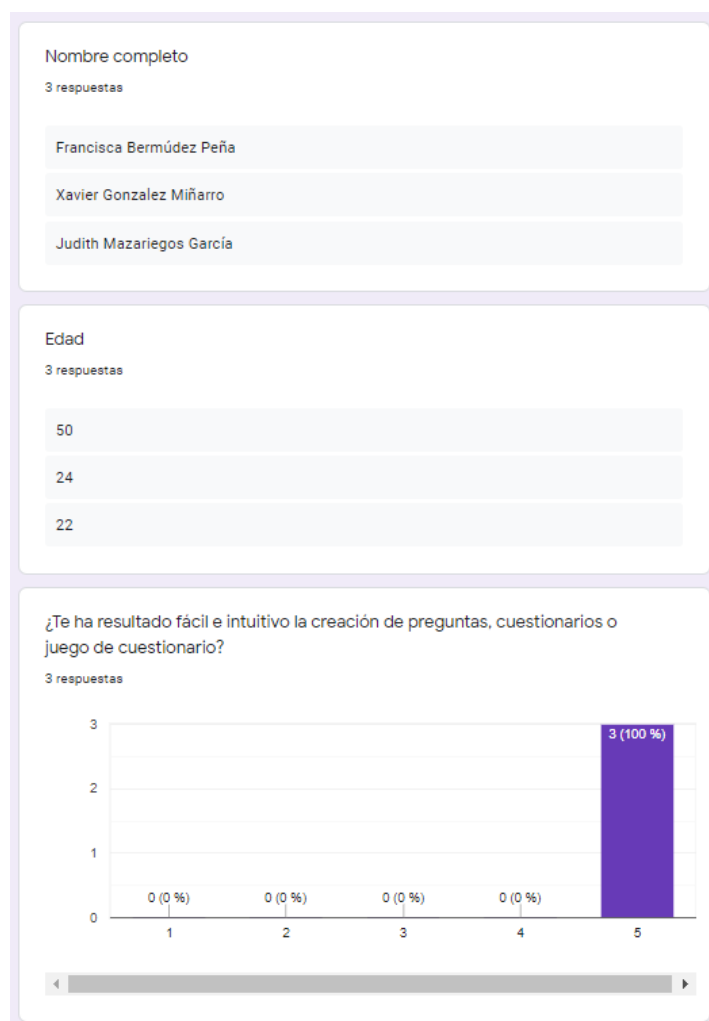
**Figura 10.1** Formulario de evaluación del proyecto

Cabe decir que me hubiera gustado probar y evaluar esta aplicación en un colegio con un entorno docente adecuado, para de esta forma poder sacar unos resultados aún más reales, pero por la situación que estamos sufriendo actualmente (la pandemia de COVID-19) me ha sido imposible poder realizar estas pruebas y me he tenido que adaptar a la situación. También es cierto que las personas que me han ayudado a evaluarlo son de mi completa confianza y están muy capacitadas para darme un buen feedback.

Las personas que van a probar la aplicación son:

- ❖ Francisca Bermudez Peña, mi madre, que es técnica de software en la empresa Sage Logic Control y por lo tanto está dentro del mundo de la tecnología y las aplicaciones.
- ❖ Judith Mazariegos García, una buena amiga, que está acabando la carrera de Biología y le encanta todo el mundo de la creación y desarrollo de aplicaciones.
- ❖ Xavier González Miñarro, un buen amigo, que está acabando la carrera de diseño industrial, le apasionan las aplicaciones móviles y, por otro lado, es director de un grupo de jóvenes (es un grupo con el que se hacen actividades educativas todos los sábados).

Sus respuestas al formulario son las siguientes (**figura 10.2**):



¿Harias alguna modificación?

3 respuestas

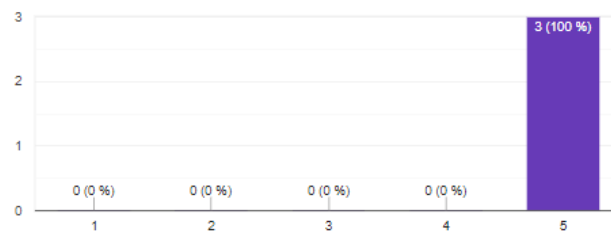
No, me ha parecido fácil e intuitivo

El problema que detecto es que para crear un juego cuestionario, la cantidad de pasos es muy elevada y cuantos menos pasos para realizar una acción, mucho mejor.

Creo que debería haber la opción de que el profesor escoja el número de respuestas para cada pregunta. Faltan alertas sobre la puntuación al escribir números con decimales.

¿Te ha resultado fácil e intuitivo la modificación de preguntas, cuestionarios o juego de cuestionario?

3 respuestas



¿Harias alguna modificación?

3 respuestas

Todo correcto

No, todo lo necesario para editar está accesible

No.

¿El modelo de pregunta te ha parecido correcto? ¿Harias algún cambio?

3 respuestas

Permitiría que el profesor pueda decidir el número de preguntas correctas e incorrectas

Estaría bien que el docente pudiera editar la cantidad de respuestas que va a dar por pregunta.

Poder añadir fotografías o links de vídeos.

¿Consideras que el modelo de cuestionario o juego de cuestionario contiene todo lo necesario?

2 respuestas

Si, creo que tiene todo lo necesario

En el caso del juego de cuestionario, al mostrar la información de uno ya creado hay dos campos que están descritos con terminología de programación. Sería conveniente no transmitir ese input al docente o transmitirlo con otra terminología.

Si, no creo que haga falta nada mas



**Figura 10.2** Resultados formulario

## 10.2 Evaluación de las respuestas

Después recoger y analizar cada una de las respuestas, se puede decir que he acabado muy satisfecho ya que las críticas han sido muy positivas a cerca de mi trabajo y de la aplicación en general. Se que aún tiene que mejorar y que esto solo es una primera versión de los módulos, los cuales considero son una base sólida para ir perfeccionando y añadiendo las mejoras que sean oportunas. Por ejemplo, una de las observaciones que más ha predominado dentro de las respuestas del formulario es la posibilidad de que el docente establezca el número de respuestas correctas e incorrectas de cada una de las preguntas y que por lo tanto no tenga un número fijo, o incluso que pueda añadir imágenes, videos o cualquier contenido que pueda complementar la pregunta.

Por otro lado, después de recibir estos feedbacks estoy seguro de que si hubiera probado esta aplicación en un centro docente habría sacado unas respuestas similares por parte de los profesores, pero me ha faltado la visión de un grupo de alumnos que conformen una clase, para que de esta forma pudiera sacar otra visión del resultado final. De todas formas, como he dicho antes, me ha resultado muy útil por las críticas constructivas y todas las ideas de mejora que he recibido, las cuales he recogido en el apartado de propuestas de futuras mejoras que encontraremos en el siguiente capítulo.

## CAPÍTULO 11. CONCLUSIONES

En este capítulo vamos a ver las conclusiones finales del proyecto. Empezaremos viendo las conclusiones técnicas, en las cuales me centraré en los objetivos que me marqué inicialmente y los resultados finales, y seguidamente las conclusiones personales, donde veremos cómo me ha ido y que me ha aportado el proyecto. Para acabar veremos un conjunto de propuestas de mejora para futuras versiones del proyecto.

### 11.1 Conclusiones técnicas

Inicialmente el gran objetivo planteado era el de diseñar e implementar un módulo de cuestionarios en todos los proyectos posibles de la herramienta Classpip. Para poder llegar a ese objetivo los primeros pasos fueron primordiales, ya que necesitaba familiarizarme con el código, el lenguaje utilizado, la tecnología y la estructura que tenía cada uno de los proyectos, debido a que gran parte era nuevo para mí y quería empezar con buen pie. Debido a todo eso hubo un proceso inicial de aprendizaje con ayuda de tutoriales y de búsquedas en internet, para que llegado el momento pudiera ponerme a desarrollar de una forma más sencilla y entendiendo el funcionamiento de cada proyecto y su estructura.

Previamente a la implementación me planteé un diseño mental sobre cómo iba a estar estructurado mi proyecto y pensé las diferentes funcionalidades que podría contener. Después de analizar cada una de estas ideas, con la ayuda de los tutores decidimos que funcionalidades implementaría y cuales se descartarían por el momento. Posteriormente realicé una serie de diseños para poder tener una idea clara de por donde tirar, siempre respetando los estilos gráficos de la aplicación ya que si los diseños se asemejaban a esos estilos posteriormente me sería más sencillo implementarlo. Por último, antes de empezar traté de entender el código que había implementado para que cuando yo empezara a añadir mi parte todo siguiera la misma estructura y las modificaciones que hiciera sobre el código existente no modificaran nada de lo que ya había operativo.

Seguidamente empezó el desarrollo del módulo, primero dejé lista toda la estructura de datos dentro de Classpip-services para poder trabajar con más comodidad en el resto de proyectos. Después empecé en el Dashboard creando el módulo de preguntas, a continuación el de cuestionarios y finalmente el juego de cuestionario. Hasta que no estuve completamente seguro de que había plasmado mi idea en el Dashboard no pasé a desarrollar la parte del móvil del alumno. Posiblemente la aplicación móvil ha sido la parte que me ha resultado más complicada debido a que su implementación venía completamente de cero y no tenía donde reflejarme.

Llegado a este punto puedo decir que los objetivos que me planteé inicialmente tanto en tiempo como en funcionalidades se han cumplido, ya que actualmente Classpip cuenta con un módulo de preguntas el cual tiene infinidad de utilidades, también el módulo de cuestionarios el cual otorga mucha riqueza a la

herramienta, y el juego de cuestionario el cual es otro gran valor y da mucho juego dentro de Classpip. Por otro lado, tenemos el móvil del alumno en el cual podemos disfrutar de la realización de los cuestionarios creados por el profesor y es otro de los puntos fuertes del trabajo.

## 11.2 Conclusiones personales

Echando la vista atrás, en el momento que empecé a buscar ideas de trabajo final quería algo que tuviera programación, que me llenara y que fuera un desafío para mí. Cuando me plantearon la idea de este trabajo me cautivó, ya que contiene todo lo que yo buscaba y sabía que iba a disfrutar haciéndolo. Y sin pensarlo, de repente se convirtió en mi objetivo personal.

Si que es cierto que durante el Grado de Telemática he tocado diferentes asignaturas de programación, lo cual considero que es una gran ventaja ya que, por ejemplo, los diferentes frameworks del proyecto como Angular e Ionic ya los conocía y no he empezado de 0. Pero esto no quiere decir que no haya tenido que trabajar duro, ya que he tenido que refrescar toda la información y he aprendido muchísimas cosas nuevas durante el tiempo de desarrollo de la aplicación. Considero que estas tecnologías están actualmente en auge y poder profundizar en cada una de ellas me ha otorgado una cantidad de conocimientos que ni yo me imaginaba y los cuales estoy seguro de que en el mundo laboral me van a venir muy bien.

En cuanto al concepto de gamificación, es algo que desde el primer momento llamó mi atención, porque yo soy monitor de esplai (de un grupo de jóvenes para los cuales monto actividades todos los sábados por la tarde) y, por lo tanto, todo el tema relacionado con jóvenes me atrae, y pensar en poder desarrollar una herramienta que llame su atención, mejore su motivación y les ayude en el día a día en la escuela, pues me llenaba mucho a nivel personal y lo vi un muy buen punto de partida.

Finalmente cabe decir que estoy muy satisfecho del resultado final del trabajo, ya que los objetivos que me marque inicialmente considero que los he cumplido y que hay operativo un buen módulo en Classpip, además haberme enfrentado a este reto yo solo me ha hecho mejorar como ingeniero y he me he demostrado a mí mismo que lo que me propongo lo puedo conseguir. Si es cierto que el módulo se puede mejorar, añadiendo modificaciones para que cada vez sea más completo, pero por el momento hay algo operativo y completamente funcional. Por otro lado, estoy muy orgulloso de formar parte de Classpip y de pensar que en el momento que salga al mercado, yo habré formado parte de ello y aunque sea en una pequeña escala también es mi proyecto.

## 11.3 Propuestas para futuras mejoras

Para finalizar aquí tenemos algunas mejoras posibles, de cara a futuras versiones:

- Dar opción al docente de establecer el número de respuestas correctas e incorrectas de una pregunta y que no sea un número fijo.
- Crear nuevos tipos de preguntas dentro de un cuestionario, como por ejemplo: de relacionar, verdadero o falso, con imagen o algún video, de escribir ...
- Implementar una marca temporal para el momento en el cual se empieza el cuestionario y para cuando se finaliza. De esta forma veríamos el tiempo que ha tardado cada uno de los alumnos en realizarlo y también nos daría opción a desempatar en la clasificación entre dos alumnos con la misma nota, ya que el que menos tarde quedaría por delante del otro.
- Encontrar la manera de que el profesor decida en que posición estará la respuesta correcta dentro de un cuestionario con la misma ordenación para todo el mundo, ya que por el momento tienen una posición fija y tal vez sería más oportuno que ese aspecto lo decidiera el profesor.
- Implementar una forma de procesar todas las respuestas de las preguntas y por lo tanto de los cuestionarios, ya que es una información muy interesante de gestionar de cara a extraer resultados o conclusiones de cómo ha ido un cuestionario o de cómo va el alumno en un momento determinado del curso.



## ANEXOS

### ANEXO 1 Resultado final

Una vez finalizado el desarrollo de cada uno de los módulos vamos a ver el resultado. Cabe decir que cada una de las pantallas ha estado diseñada intentado asemejarse lo máximo posible a las pantallas que he diseñado anteriormente que tenemos en el capítulo 6.

Empezaremos con el Dashboard:

#### Pantallas Dashboard

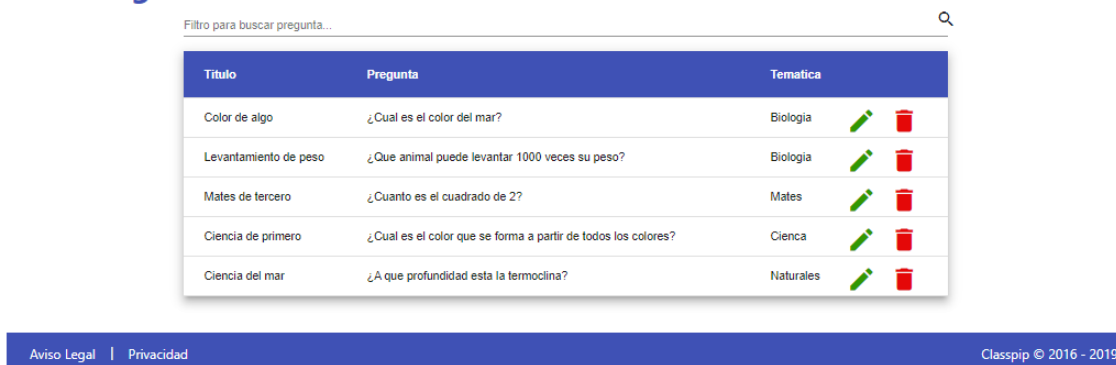
A continuación, veremos un seguido de capturas de pantalla del resultado final de cada uno de los módulos implementados en el Dashboard.

Empezaremos por el módulo de preguntas.

#### Módulo de preguntas

Si nos fijamos en la **figura 1.1** podremos ver resultado de la pantalla de “Mis Preguntas” en la cual podremos hacer una búsqueda de la pregunta que deseemos ver/editar (**figura 1.2**) o eliminar. Seguidamente, desde la **figura 1.3** a la **figura 1.5** veremos el proceso de creación de preguntas.

#### Mis Preguntas



Titulo	Pregunta	Temática
Color de algo	¿Cual es el color del mar?	Biología
Levantamiento de peso	¿Que animal puede levantar 1000 veces su peso?	Biología
Mates de tercero	¿Cuanto es el cuadrado de 2?	Mates
Ciencia de primero	¿Cual es el color que se forma a partir de todos los colores?	Ciencia
Ciencia del mar	¿A que profundidad esta la termoclina?	Naturales

**Figura 1.1** Mis preguntas

### Editar pregunta

Título \*

Levantamiento de peso

Pregunta \*

¿Que animal puede levantar 1000 veces su peso?

Temática \*

Biología

Respuesta Correcta \*

Hormiga

Respuesta Incorrecta 1 \*

Elefante

Respuesta Incorrecta 2 \*

Girafa

Respuesta Incorrecta 3 \*

Tortuga

Feedback Correcto \*

Bien

Feedback Incorrecto \*

Mal

Cerrar Guardar

**Figura 1.2** Editar pregunta

1 Pregunta 2 Añadir Respuestas 3 Finalizar

Título \*

Ciencia del mar

Pregunta \*

¿A que profundidad esta la termoclina?

Temática \*

Naturales

Atrás Siguiente

**Figura 1.3** Crear pregunta

1 Pregunta 2 Añadir Respuestas 3 Finalizar

Respuesta Correcta \*

200-1000m

Respuesta Incorrecta 1 \*

100-200m

Respuesta Incorrecta 2 \*

50-5000m

Respuesta Incorrecta 3 \*

10-10m

Feedback Correcto \*

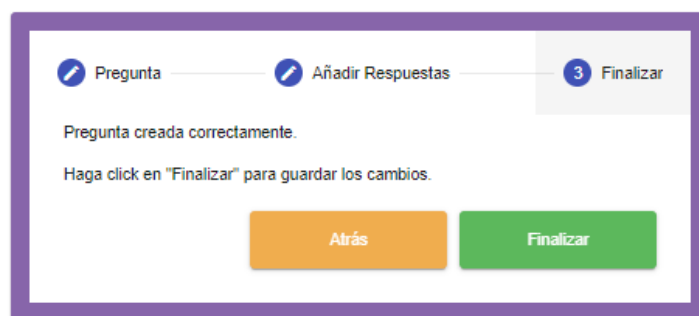
Muy bien

Feedback Incorrecto \*

Mira el capítulo 2 del libro

Atrás Siguiente

**Figura 1.4** Crear pregunta 2
























**Figura 1.5** Crear pregunta 3

### ***Módulo de cuestionarios***

La primera imagen (**figura 1.6**) nos muestra el resultado de “Mis Cuestionarios”, en el cual como pasa en el caso de las preguntas, podremos hacer una búsqueda del cuestionario que queramos y podremos ver/editar (**figura 1.7**) o eliminar. Seguidamente, desde la **figura 1.8** a la **figura 1.10** veremos el proceso de creación de cuestionarios.

#### **Mis Cuestionarios**

Filtro para buscar cuestionario... 

Titulo	Descripcion		
El mas chungo	Aprueballo		
Mates de tercero	Si podeis realizar el siguiente cuestionario aprobais		
Cuestionario final	Si sacáis un 6 o mas aprobáis la asignatura		
Test	Test		
Filósofos del mundo	Filosofía		
Cuestionario mas importante	Acertrar todo		
Prueba	Prueba2		
Ciencias del mundo	¿Que es lo que sabes?		
Mates tema 3	Si habéis estudiado sacareis nota		

Aviso Legal | Privacidad Classpip © 2016 - 2019

**Figura 1.6** Mis cuestionarios

## Editar Cuestionario

**Título:** El mas chungo

**Descripción:** Apruebalo para poder conseguir un punto mas en la asignatura

[Aceptar Cambios ✓](#)

[Añadir/Eliminar Preguntas](#)

Filtro para buscar pregunta...

Título	Pregunta	Temática
Hola	3=3	Mates
Color de algo	¿Cual es el color del mar?	Biología
Nueva pregunta	Como me llamo	Personal

[Volver](#)

**Figura 1.7** Editar cuestionario

1 Descripción — 2 Añadir preguntas — 3 Finalizar

Título \*

Mates tema 3

Descripción \*

Si habéis estudiado sacareis nota

[Siguiente](#)

**Figura 1.8** Crear cuestionario

1 Descripción — 2 Añadir preguntas — 3 Finalizar

[Añadir preguntas](#)

[Atrás](#) [Siguiente](#)

**Figura 1.9** Crear cuestionario 2

1 Descripción — 2 Añadir preguntas — 3 Finalizar

Cuestionario creado correctamente.

Haga click en "Finalizar" para guardar los cambios.

[Atrás](#) [Finalizar](#)

**Figura 1.10** Crear cuestionario 3

## Módulo juego de cuestionario

Por último, en el Dashboard tenemos el módulo de juego de cuestionario, en el cual tenemos opción de crearlo como vemos desde la **figura 1.11** hasta la **figura 1.16** y seguidamente el profesor tendrá el juego preparado y dispondrá de la opción de activar el juego para que puedan jugar, ver la información (y editar) o eliminarlo (**figura 1.17**).

The screenshot shows the first step of a four-step process to create a quiz game. The steps are: 1. Introduce el nombre del juego, 2. Tipo De Juego, 3. Modo De Juego, and 4. Finalizar. In step 1, there is a text input field with the placeholder 'Introduce el Nombre del juego' and the text 'Cuestionario difícil' entered. A blue 'Siguiente' (Next) button is located at the bottom right.

Figura 1.11 Crear juego de cuestionario

The screenshot shows the second step of the process. The steps are: 1. Introduce el nombre del juego, 2. Tipo De Juego, 3. Cuestionario, 4. Puntuación, 5. Presentación, and 6. Finalizar. In step 2, there is a section titled 'Selecciona un tipo de juego:' with four buttons: 'Juego De Puntos' (blue), 'Juego De Colección' (pink), 'Juego De Competición' (red), and 'Juego De Cuestionario' (blue). Below the buttons, it says 'Has elegido ... Juego De Cuestionario'. At the bottom right, there are two buttons: 'Atrás' (orange) and 'Siguiente' (blue).

Figura 1.12 Crear juego de cuestionario 2

The screenshot shows the third step of the process. The steps are: 1. Introduce el nombre del juego, 2. Tipo De Juego, 3. Cuestionario, 4. Puntuación, 5. Presentación, and 6. Finalizar. In step 3, there is a blue button labeled 'Añadir cuestionario' with a plus icon. At the bottom right, there are two buttons: 'Atrás' (orange) and 'Siguiente' (blue).

Figura 1.13 Crear juego de cuestionario 3

The screenshot shows the fourth step of the process. The steps are: 1. Introduce el nombre del juego, 2. Tipo De Juego, 3. Cuestionario, 4. Puntuación, 5. Presentación, and 6. Finalizar. In step 4, there are two text input fields. The first is labeled 'Introduce la puntuación por respuesta correcta' and has the value '2' entered. The second is labeled 'Introduce la puntuación por respuesta incorrecta (si no quiere restar ponga 0)' and has the value '1' entered. At the bottom right, there are two buttons: 'Atrás' (orange) and 'Siguiente' (blue).

Figura 1.14 Crear juego de cuestionario 4

**Figura 1.15** Crear juego de cuestionario 5

**Figura 1.16** Crear juego de cuestionario 6

## Juego De Cuestionario

Información ⓘ
Activar ✓
Eliminar ✖

Filtro para buscar pregunta... 🔍

Nombre	Primer Apellido	Segundo Apellido	Nota
Pol	Peinado	Alcaide	0.1
Sergio	Sánchez	Plaza	0
Albert	Lillo	Méndez	0
Noa	Garcia	Esteve	0
Berta	Junqué	Llaudet	0
Victor	Pérez	Batlle	0

Volver

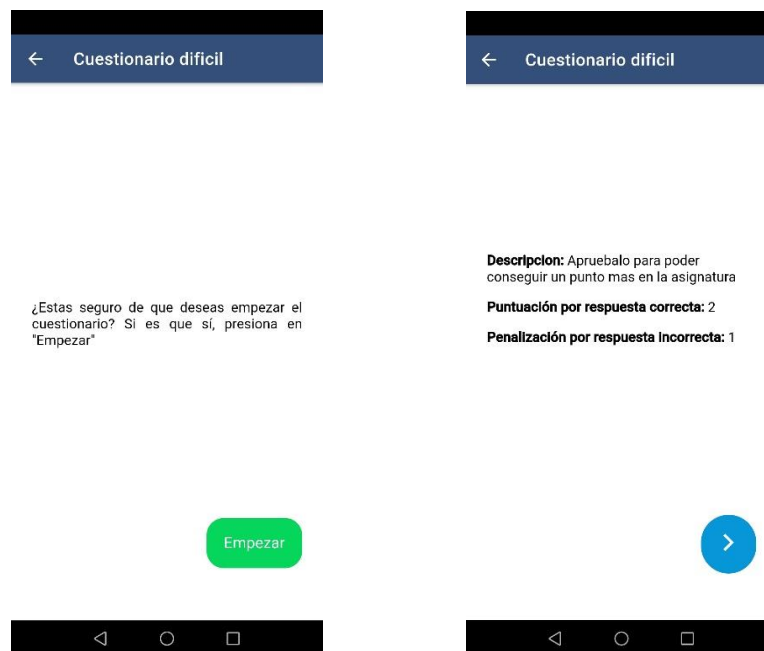
**Figura 1.17** Información juego de cuestionario activo

## Pantallas Mobile-Alumno

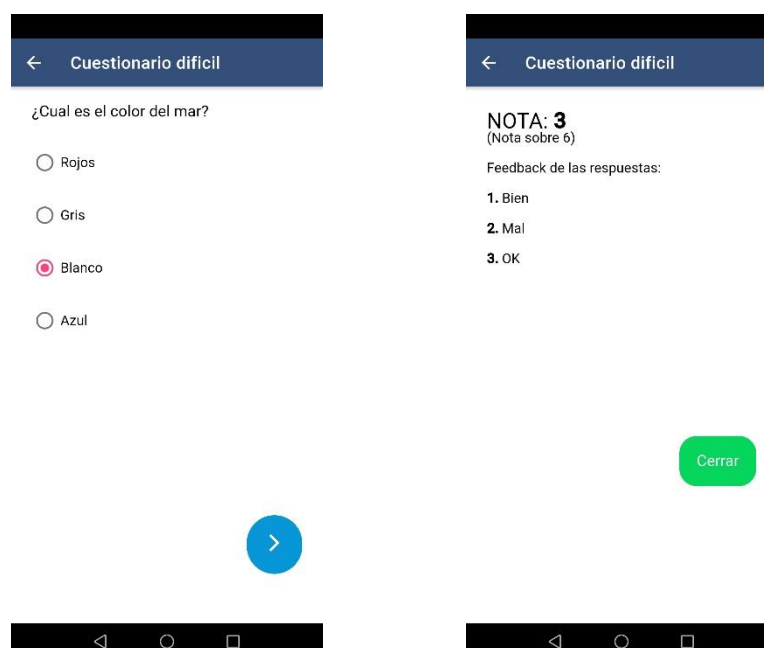
En este punto vamos a ver el resultado de las pantallas diseñadas y por lo tanto operativas del móvil del alumno.

### *Módulo juego de cuestionario*

Primero vamos a ver el proceso de realización del juego de cuestionario por parte del alumno (**figura 1.18** y **figura 1.19**) empezando por la pantalla inicial y terminando en la pantalla del feedback del examen.

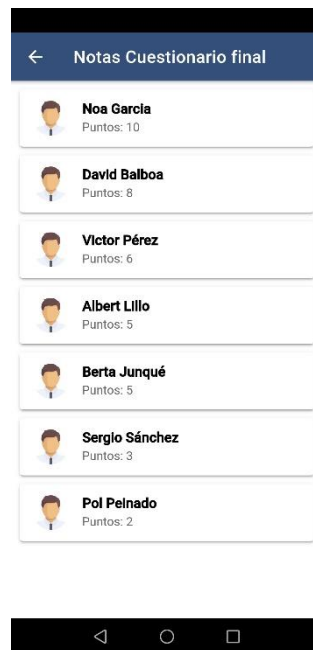









**Figura 1.18** Realización de un cuestionario paso 1 y 2



**Figura 1.19** Realización del cuestionario y feedback final

Por último, en la **figura 1.20** podemos ver la tabla de alumnos que han realizado el cuestionario una vez ya finalizado ordenada de mejor a peor nota.



← Notas Cuestionario final	
	<b>Noa García</b> Puntos: 10
	<b>David Balboa</b> Puntos: 8
	<b>Victor Pérez</b> Puntos: 6
	<b>Albert Lillo</b> Puntos: 5
	<b>Berta Junqué</b> Puntos: 5
	<b>Sergio Sánchez</b> Puntos: 3
	<b>Pol Pelnado</b> Puntos: 2

**Figura 1.20** Notas de los alumnos en un juego de cuestionario finalizado



## ANEXO 2 Estilos gráficos

Los estilos gráficos del proyecto son una parte fundamental ya que nos permiten de una manera gráfica agregar información visual y son muy importantes en la interacción con el usuario. Los diferentes estilos de esta aplicación están hechos de tal forma que sean amigables e intuitivos para el usuario. En mi caso, no he tenido que escoger ningún estilo gráfico debido a que he seguido con lo que ya había implementado dentro de los proyectos para que todo siga el mismo rumbo.

Debido a que he trabajado en dos proyectos distintos voy a separar los estilos en ellos dos:

### Estilos gráficos Dashboard

#### *Títulos*

Cada una de las pantallas de los componentes tiene un título como mínimo y alguna de las pantallas también tiene subtítulo. Para cada uno de ellos he utilizado el mismo css que vemos en la **figura 2.1** para que tengan el mismo formato y no cambien unos a otros.

```
.titulo{
  width: 70%;
  margin-left: 15%;
  height: 60px;
  padding: 10px;
  color: rgb(63, 81, 181);
  font-size: 2.5rem !important;
  font-weight: bold;
}

.subtitulo{
  width: 70%;
  margin-left: 15%;
  height: 45px;
  padding: 10px;
  color: rgb(63, 81, 181);
  font-size: 1.5rem !important;
}
```

**Figura 2.1** Estilo titulo y subtitulo

#### *Botones*

En este caso he utilizado diferentes tipos de botones, pero todos y cada uno de ellos están en el css global de la aplicación, y tenemos unos ejemplos en la **figura 2.2** en los cuales podemos ver que hay diferentes colores en función de la utilidad del botón.

```
.btn.Agregar{
  background-color: rgb(65, 178, 212);
}
.btn.Info{
  background-color: rgb(240, 173, 78);
}
.btn.Back{
  background-color: rgb(240, 173, 78);
}
.btn.Aceptar{
  background-color: rgb(92, 184, 92);
}
.btn.Eliminar{
  background-color: rgb(212, 57, 52);
}
.btn.Editar{
  background-color: rgb(65, 178, 212);
}
```

**Figura 2.2** Estilos botones

## Tablas

En el caso de las tablas, para darles forma hago uso de las propiedades nativas de las tablas y por otro lado para los títulos utilizo “mat-header-row” que es lo que establece el color y también “tituloColumnaTabla” (**figura 2.3**) que define las características del texto de los títulos.

```
// Opciones gráficas tablas
.mat-header-row{
  text-align: center;
  background-color: rgb(63, 81, 181);
  // background-color: rgb(63, 181, 79);
}

.tituloColumnaTabla{
  color: white;
  text-align: left !important;
  font-weight: bold;
  font-size: 90%;
}
```

**Figura 2.3** Estilo título tablas

## Módulo de creación

Para la creación de preguntas, cuestionarios o juego de cuestionario he hecho uso de un Stepper, el cual, para tener un mejor diseño, lo encuadro dentro de un marco lila (**figura 2.4**). El resto de componentes como por ejemplo los botones utilizamos los estilos comentados anteriormente.

```
.panel{
  background-color: #8765aa;
  width: 600px;
}
```

**Figura 2.4** Marco de Stepper

## Alertas

Para el caso de las alertas (**figura 2.5**) he hecho uso de Sweet Alert 2, que es una librería la cual proporciona todo tipo de emergentes, los cuales quedan muy bien y son fáciles de utilizar.

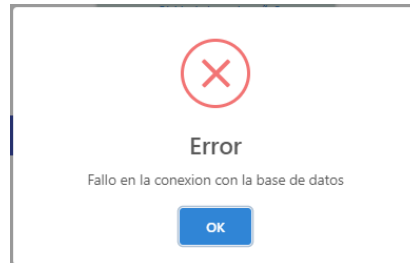


Figura 2.5 Alertas

## Estilos gráficos Mobile-Alumno

La ventaja que nos ofrece Ionic es que muchos aspectos que hemos utilizado dentro de la aplicación angular los podemos reutilizar adaptando el código de la forma correcta. Ahora veremos cómo se han aplicado los componentes dentro del código.

## Botones

Uno de los elementos más utilizados han sido los botones, los cuales he utilizado para empezar el juego, cambiar de preguntas y finalizar. El estilo que he utilizado es el que podemos ver en la **figura 2.6**, junto con dos estilos más, que son los de la disposición de los radio-button, los cuales se establecen en las posibles respuestas de un cuestionario.

```
.button-empezar {  
  background-color: #007060; color: white; font-size: 18px; padding: 20px; border-radius: 20px; position: absolute; bottom: 35px; right: 25px;  
}  
  
.siguiente {  
  background-color: #007060; color: white; font-size: 24px; padding: 20px; border-radius: 50px; position: absolute; bottom: 35px; right: 25px;  
}  
  
.anterior {  
  background-color: #007060; color: white; font-size: 24px; padding: 20px; border-radius: 50px; position: absolute; bottom: 35px; left: 25px;  
}  
  
.radio-group {  
  display: flex; flex-direction: column; margin: 15px 0;  
}  
  
.radio-button {  
  margin: 5px; color: black;  
}
```

Figura 2.6 Estilos botones móvil

## Tablas

Para las tablas no he hecho uso de una tabla como tal, he utilizado ion-cards (como podemos ver en la **figura 2.7**) ya que es algo interno de Ionic y quedaba más estético a la hora de mostrar un ranking de puntuaciones, por ejemplo.

```
<ion-card *ngFor="let alum of MisAlumnosDelJuegoDeCuestionario">
  <ion-item lines="none">
    <ion-avatar slot="start">
      
    </ion-avatar>
    <ion-label>
      <h2> <b>{{alum.Nombre}} {{alum.PrimerApellido}}</b> </h2>
      <p> Puntos: {{alum.Nota}}</p>
    </ion-label>
  </ion-item>
</ion-card>
```

**Figura 2.7** Formato tablas

## Módulo de juego

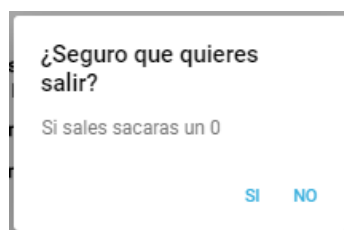
Para la realización del juego he hecho uso de algo que hemos visto antes del Dashboard, un Stepper, el cual nos ha facilitado navegar libremente entre las pantallas iniciales, preguntas y finales. La única modificación que he tenido que hacer ha sido la de ocultar la cabecera, ya que viene por defecto (**figura 2.8**).

```
:host ::ng-deep .mat-horizontal-stepper-header-container{
  display: none;
}
```

**Figura 2.8** Modificación cabecera Stepper

## Alertas

Para finalizar tenemos las alertas del móvil. Para generarlas he hecho uso de la librería alertController de Ionic y el resultado es el siguiente (**figura 2.9**).



**Figura 2.9** Alertas móvil